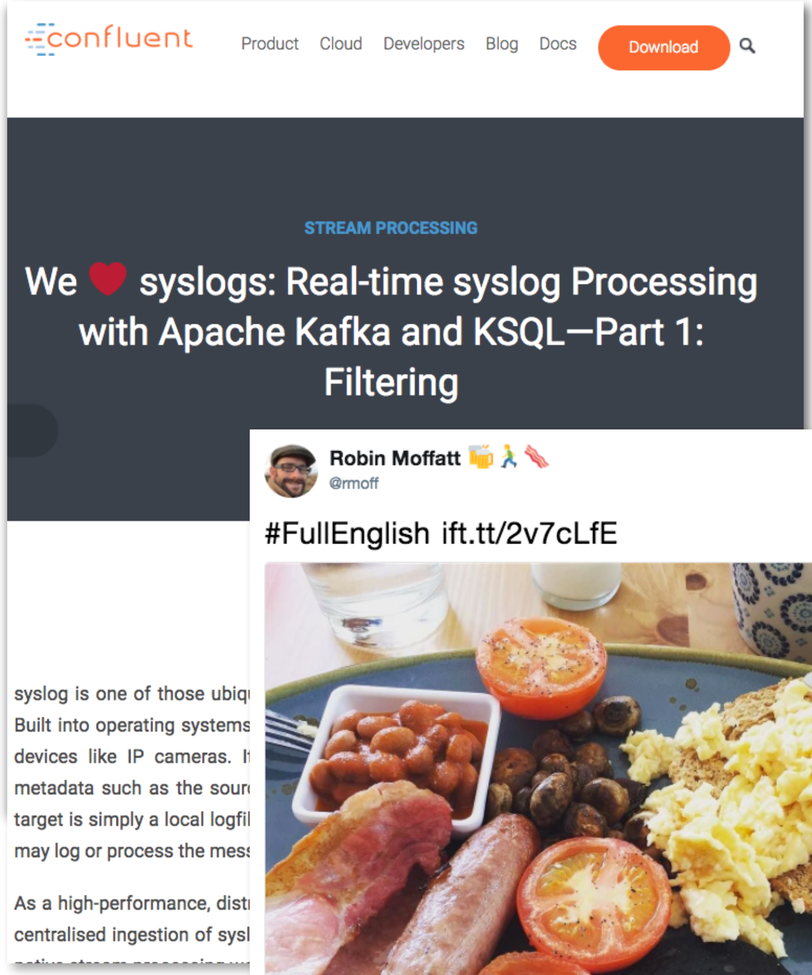
https://speakerdeck.com/rmoff/

Apache Kafka and KSQL in Action : Let's Build a Streaming Data Pipeline!

@rmoff robin@confluent.io

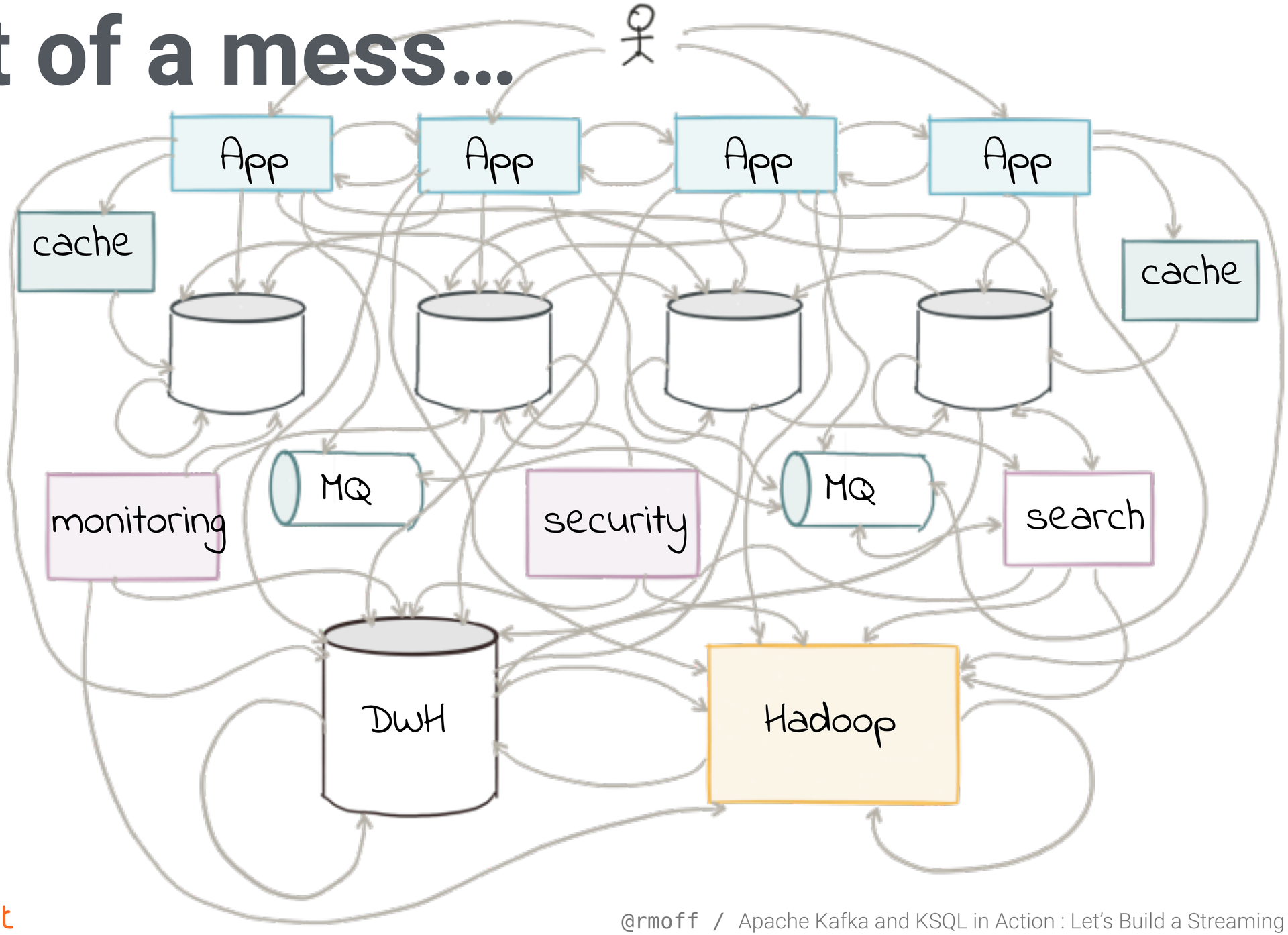
$ whoami

- Developer Advocate @ Confluent
- Working in data & analytics since 2001
- Oracle ACE Director & Dev Champion
- Blogging : http://rmoff.net & http://cnfl.io/rmoff
- Twitter: @rmoff
  - Geek stuff
  - Beer & Fried Breakfasts

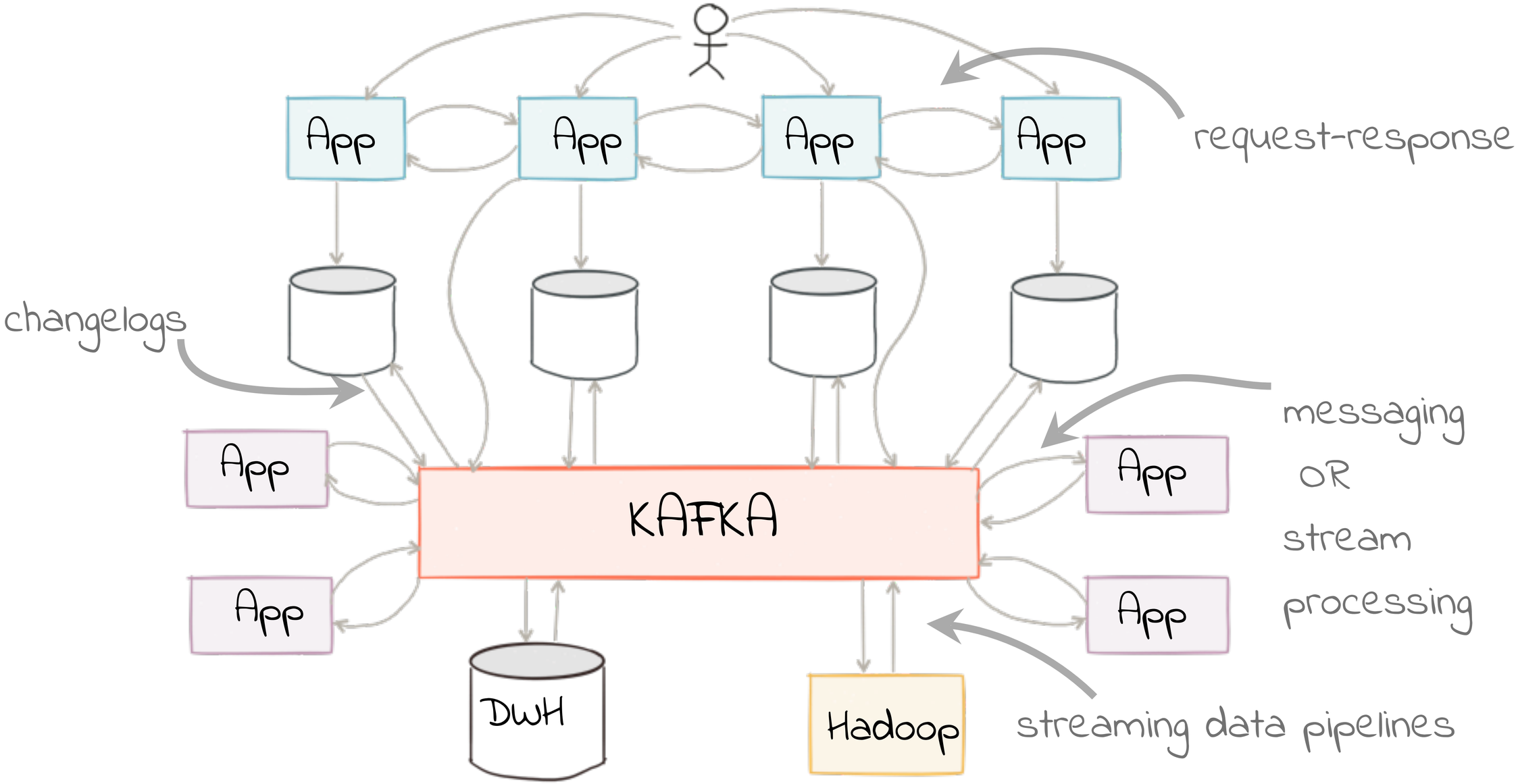


https://speakerdeck.com/rmoff/

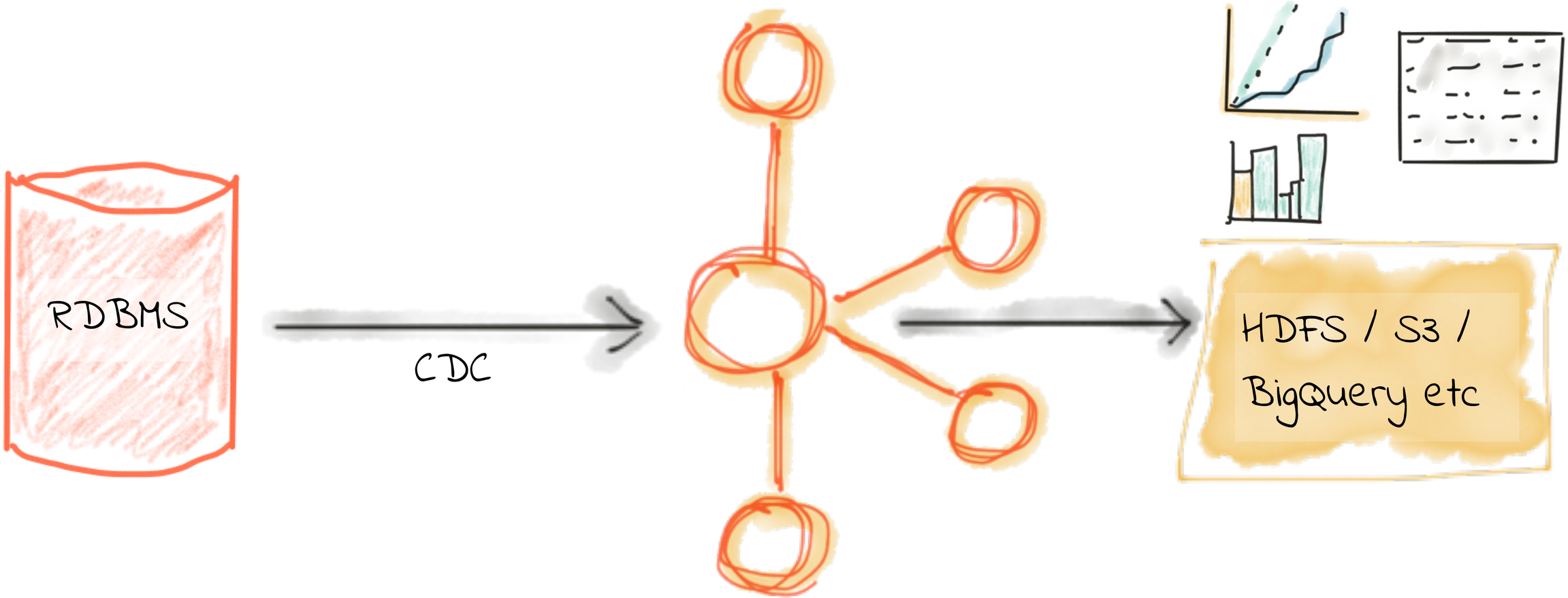
A bit of a mess…



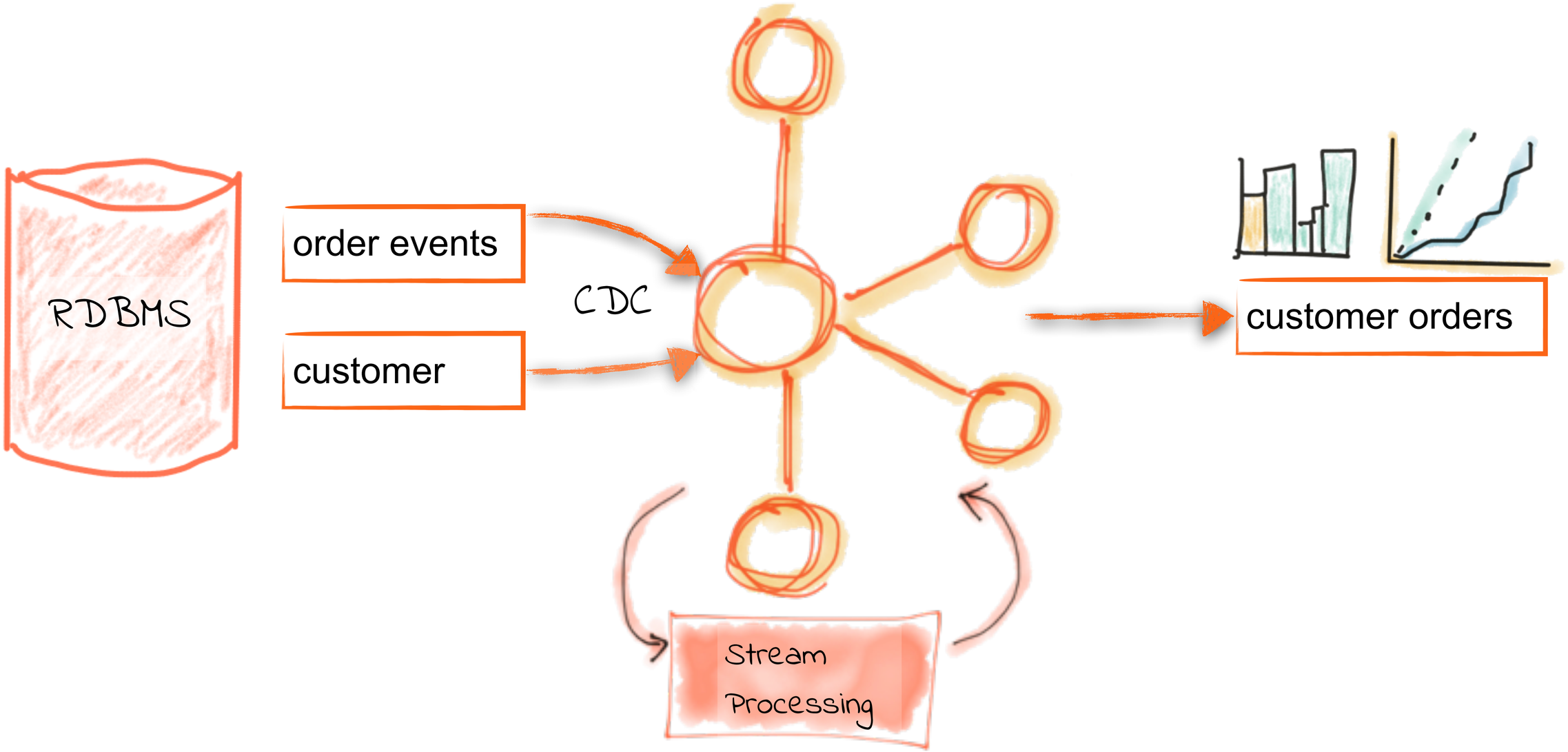
Kafka is a Streaming Platform



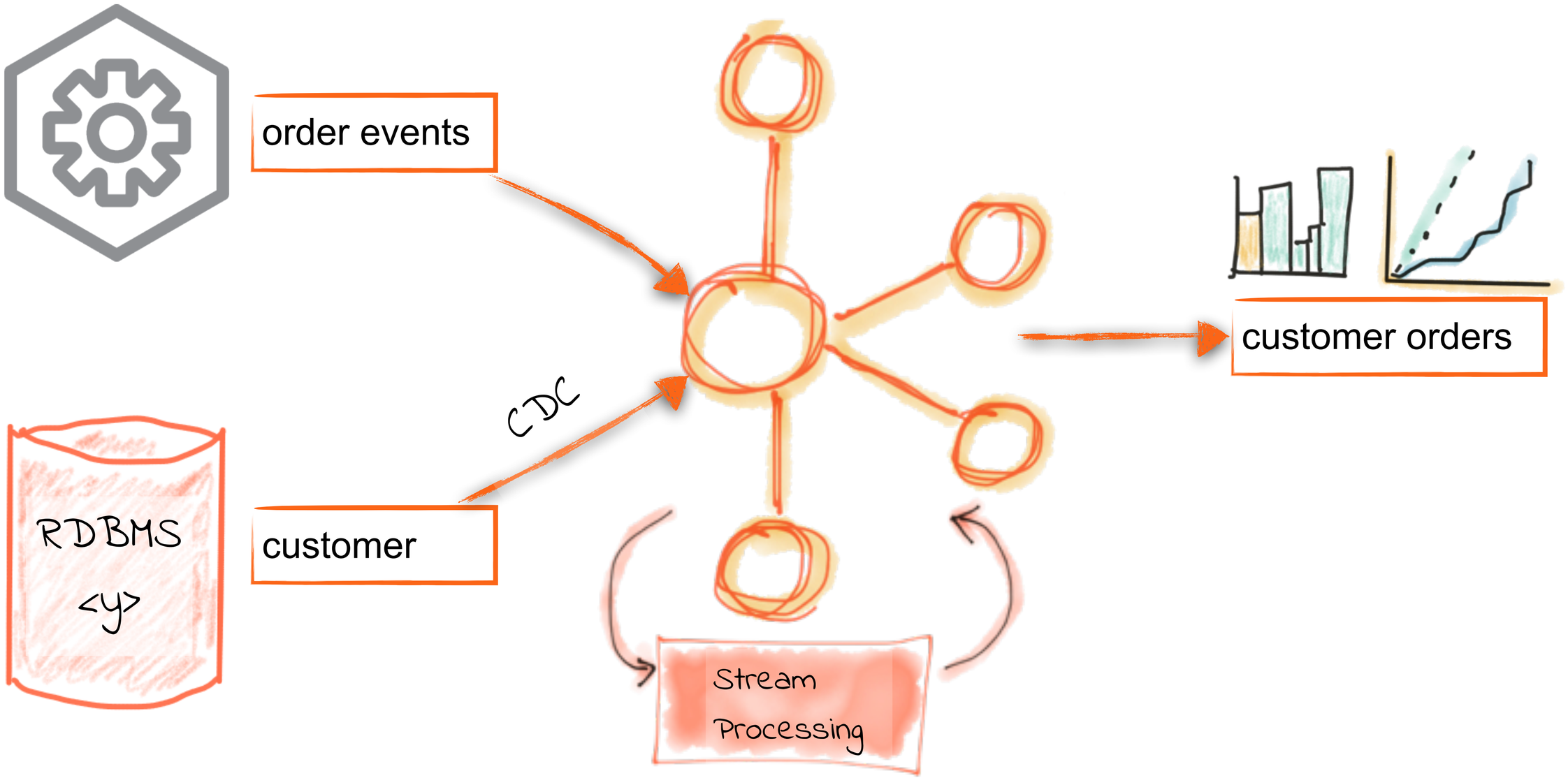
Analytics - Database Offload



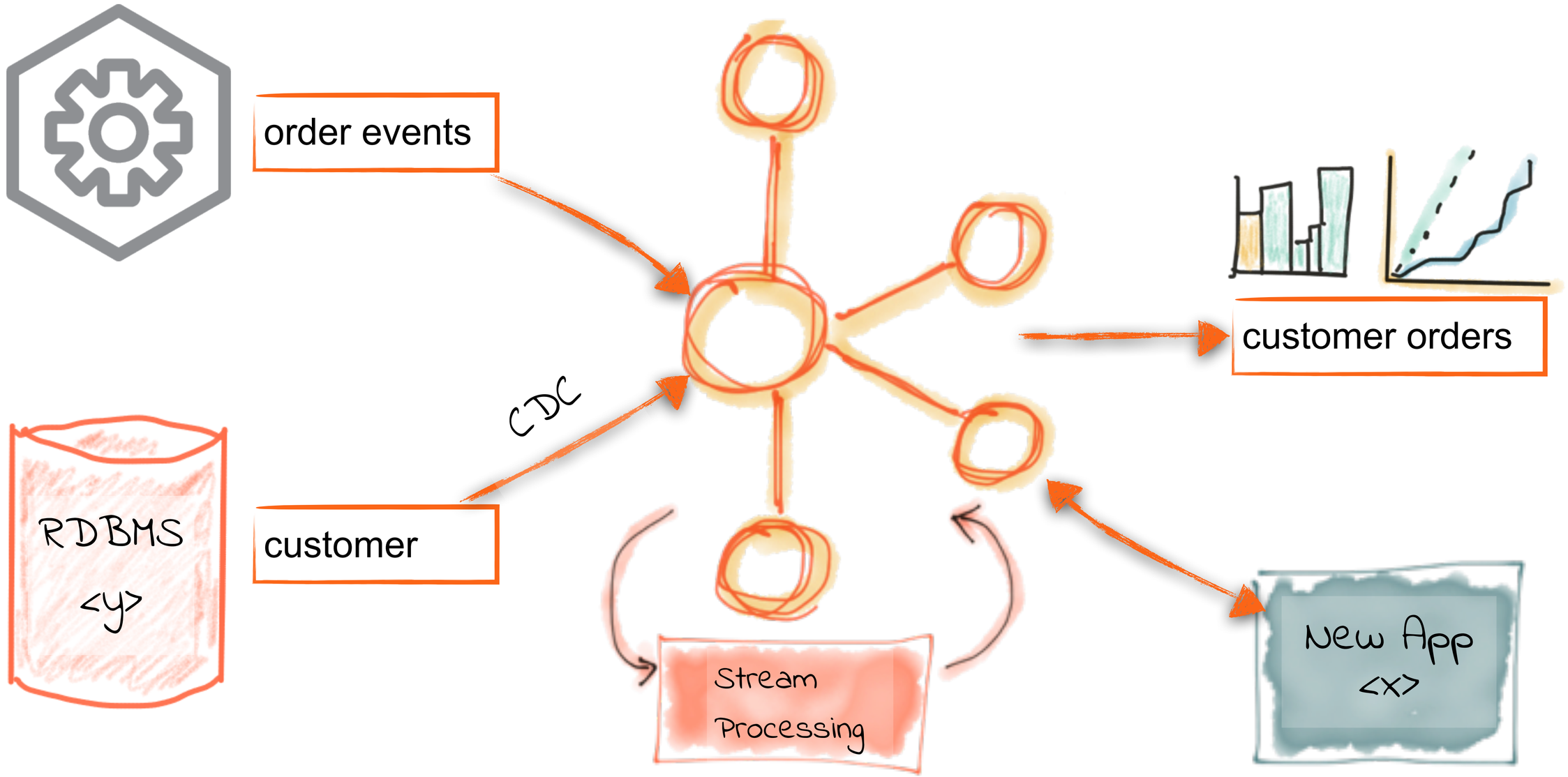
Stream Processing with Apache Kafka and KSQL



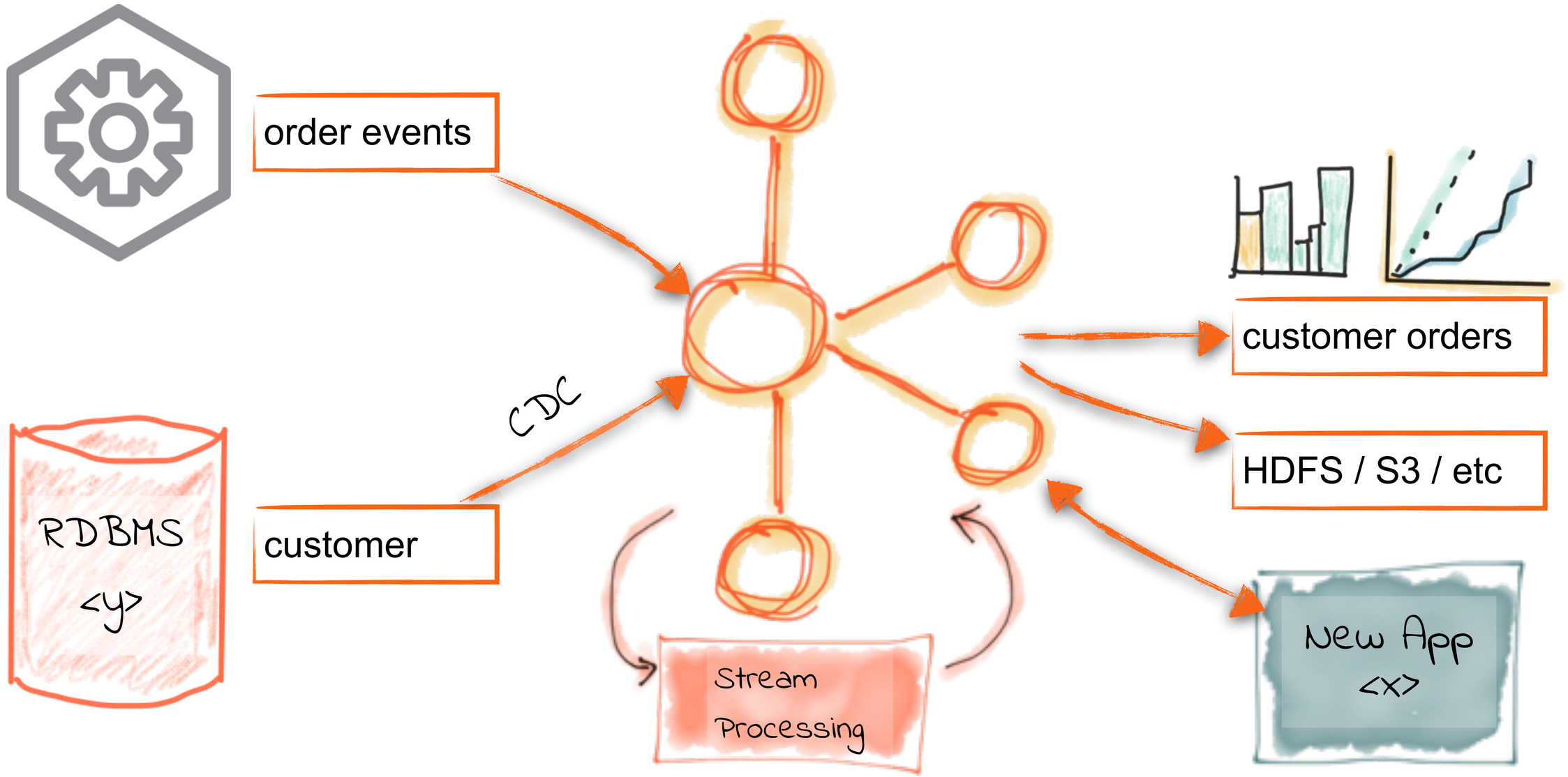
Real-time Event Stream Enrichment



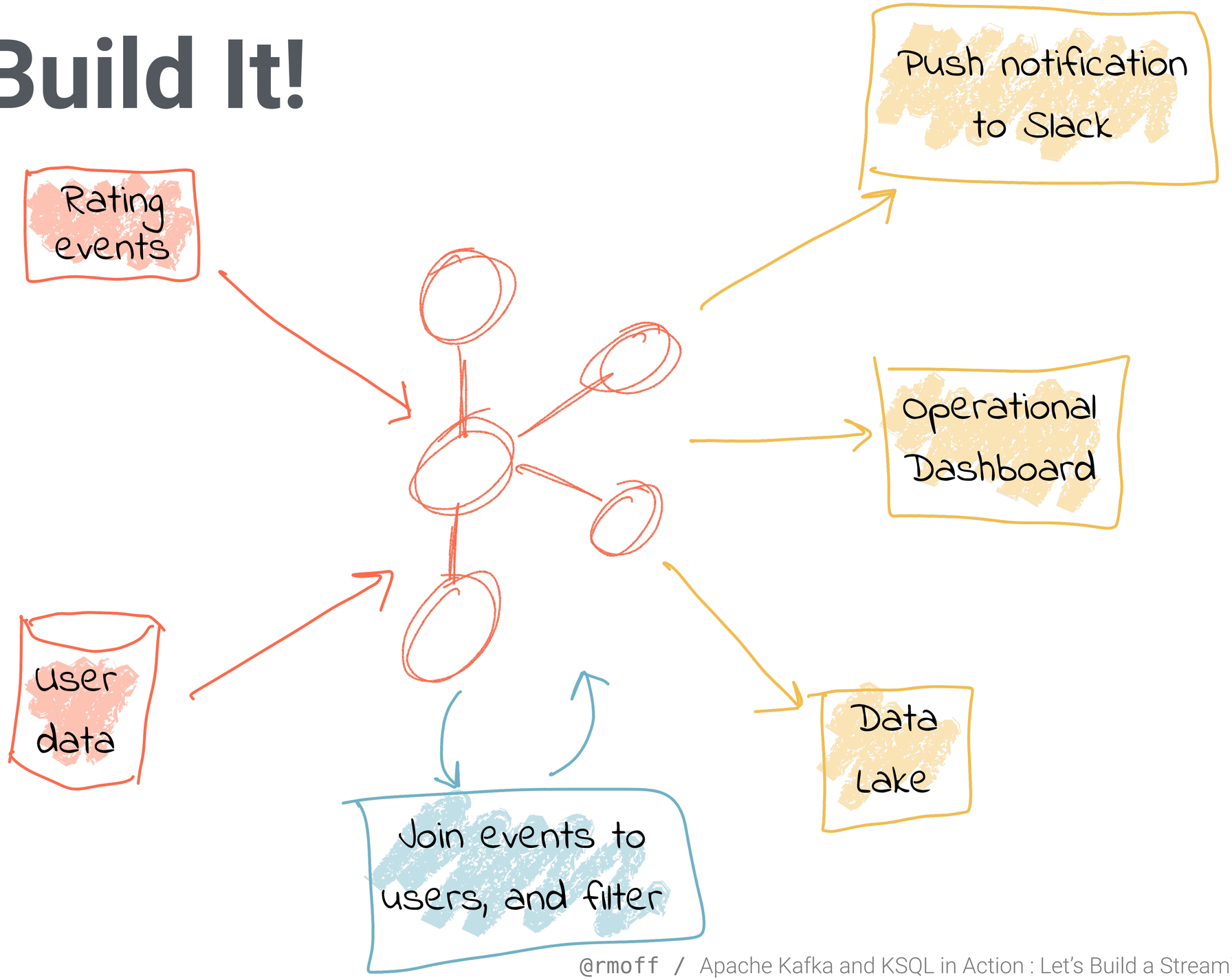
Transform Once, Use Many



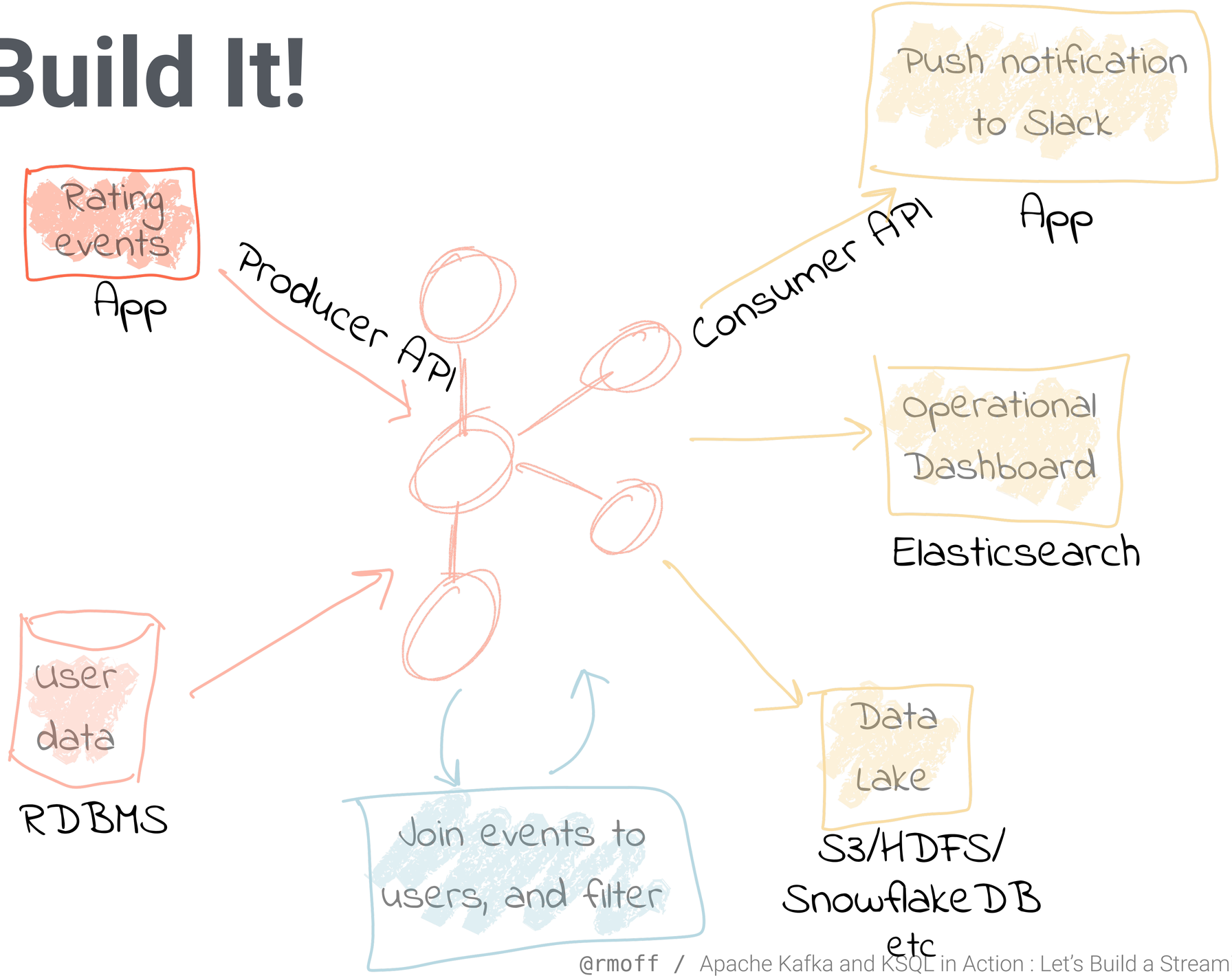
Transform Once, Use Many



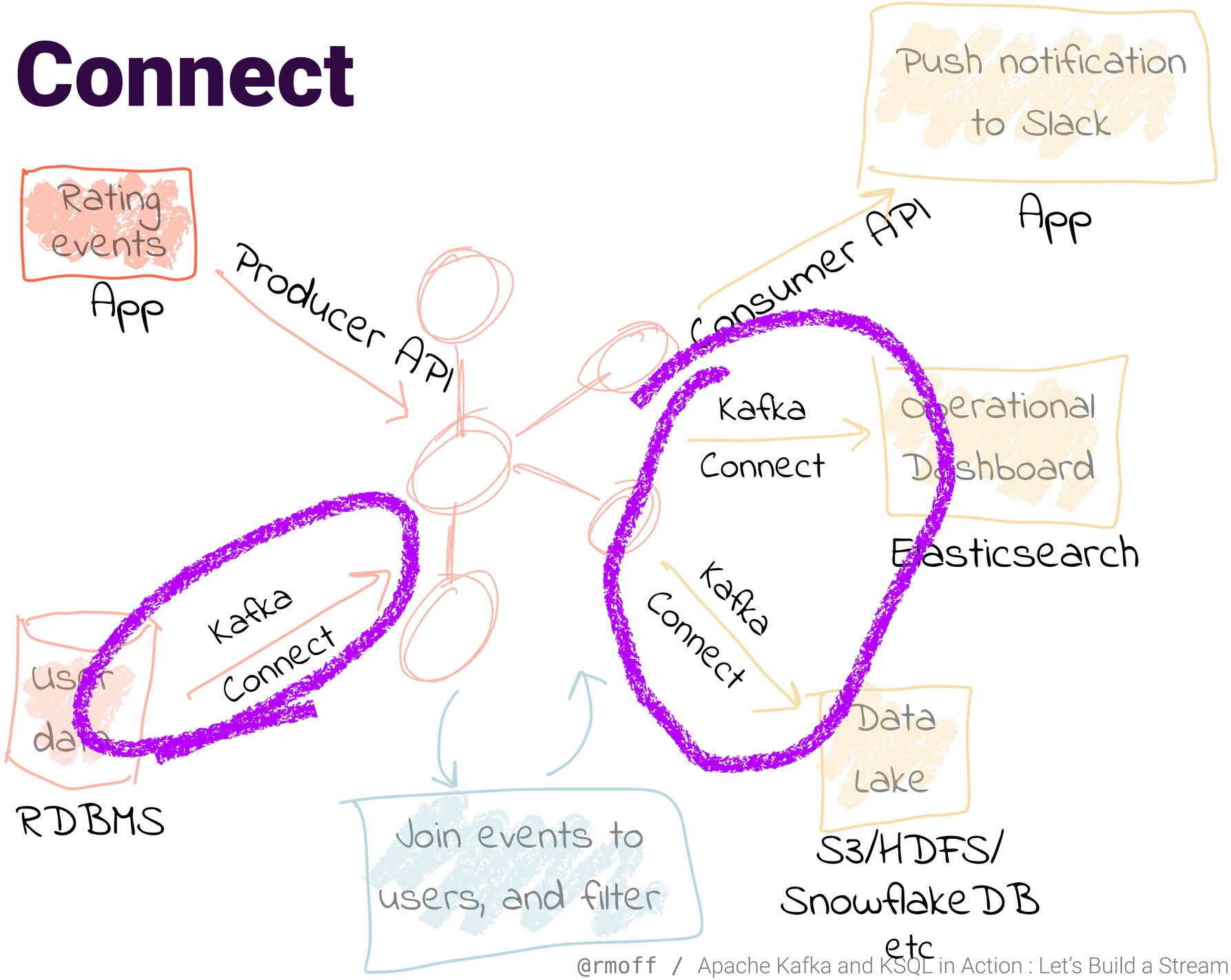
Let's Build It!



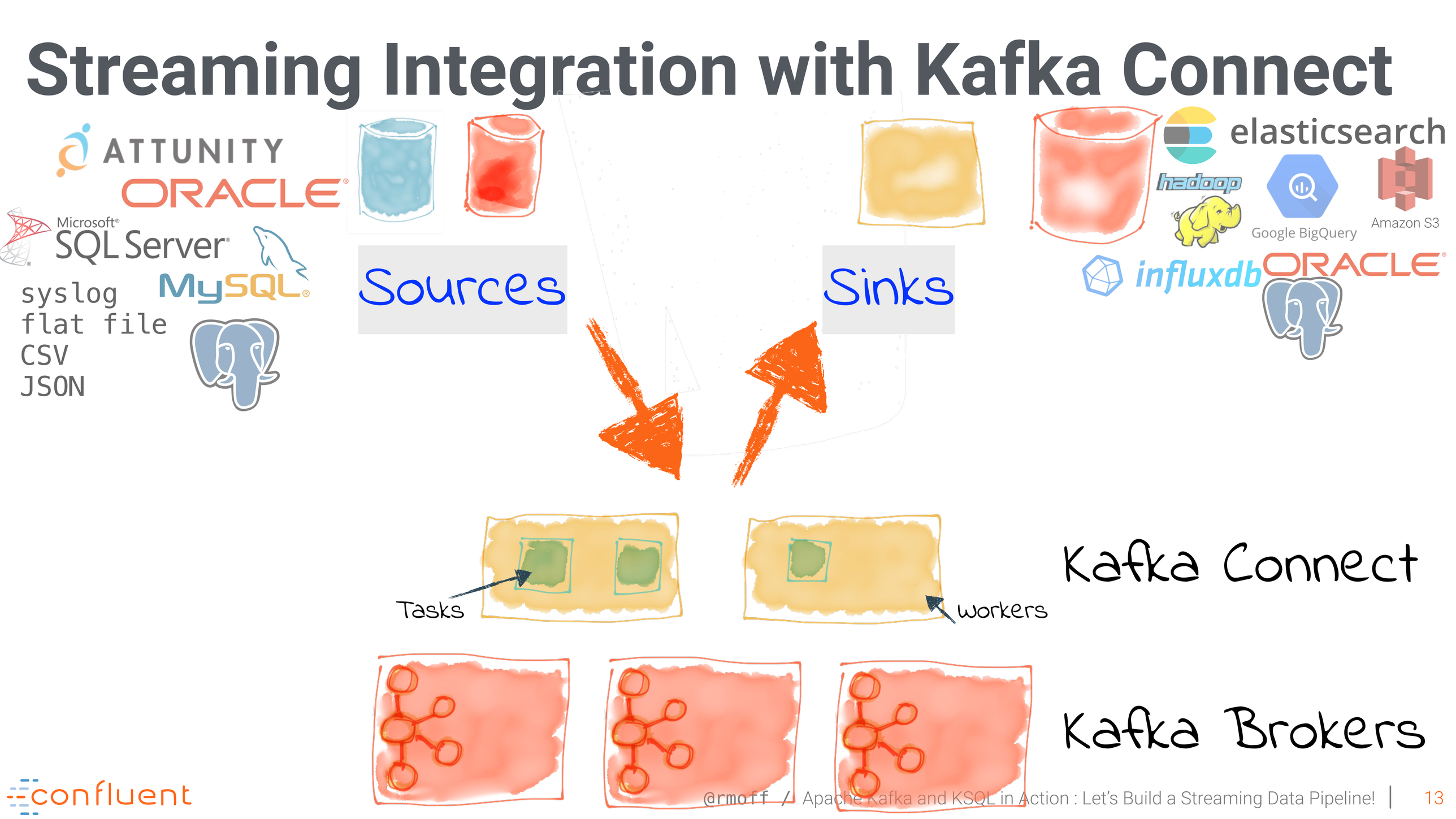
Let's Build It!



Kafka Connect



Streaming Integration with Kafka Connect



Kafka Connect

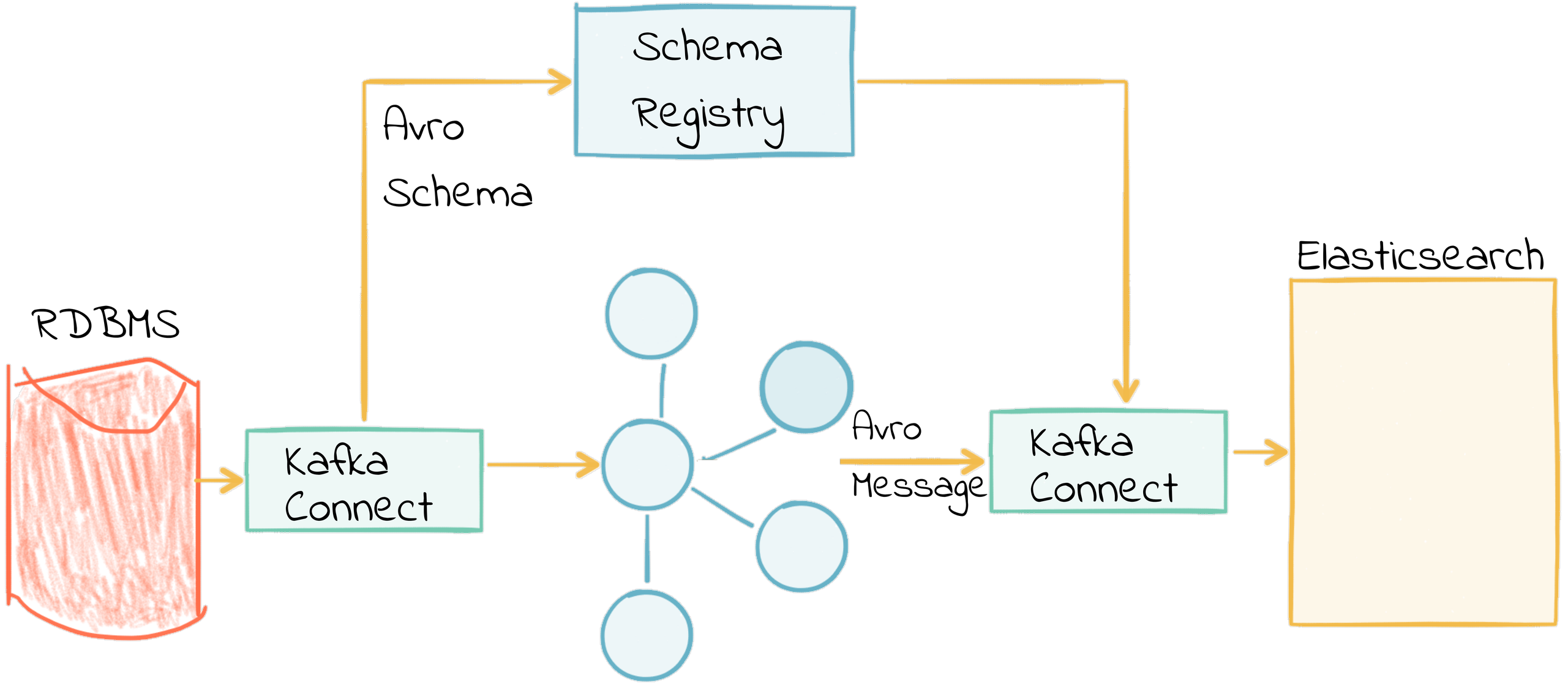
Reliable and scalable integration of Kafka with other systems – no coding required.

- ✓ Centralized management and configuration
- ✓ Support for hundreds of technologies including RDBMS, Elasticsearch, HDFS, S3
- ✓ Supports CDC ingest of events from RDBMS
- ✓ Preserves data schema
- ✓ Fault tolerant and automatically load balanced
- ✓ Extensible API
- ✓ Single Message Transforms
- ✓ Part of Apache Kafka, included in Confluent Open Source

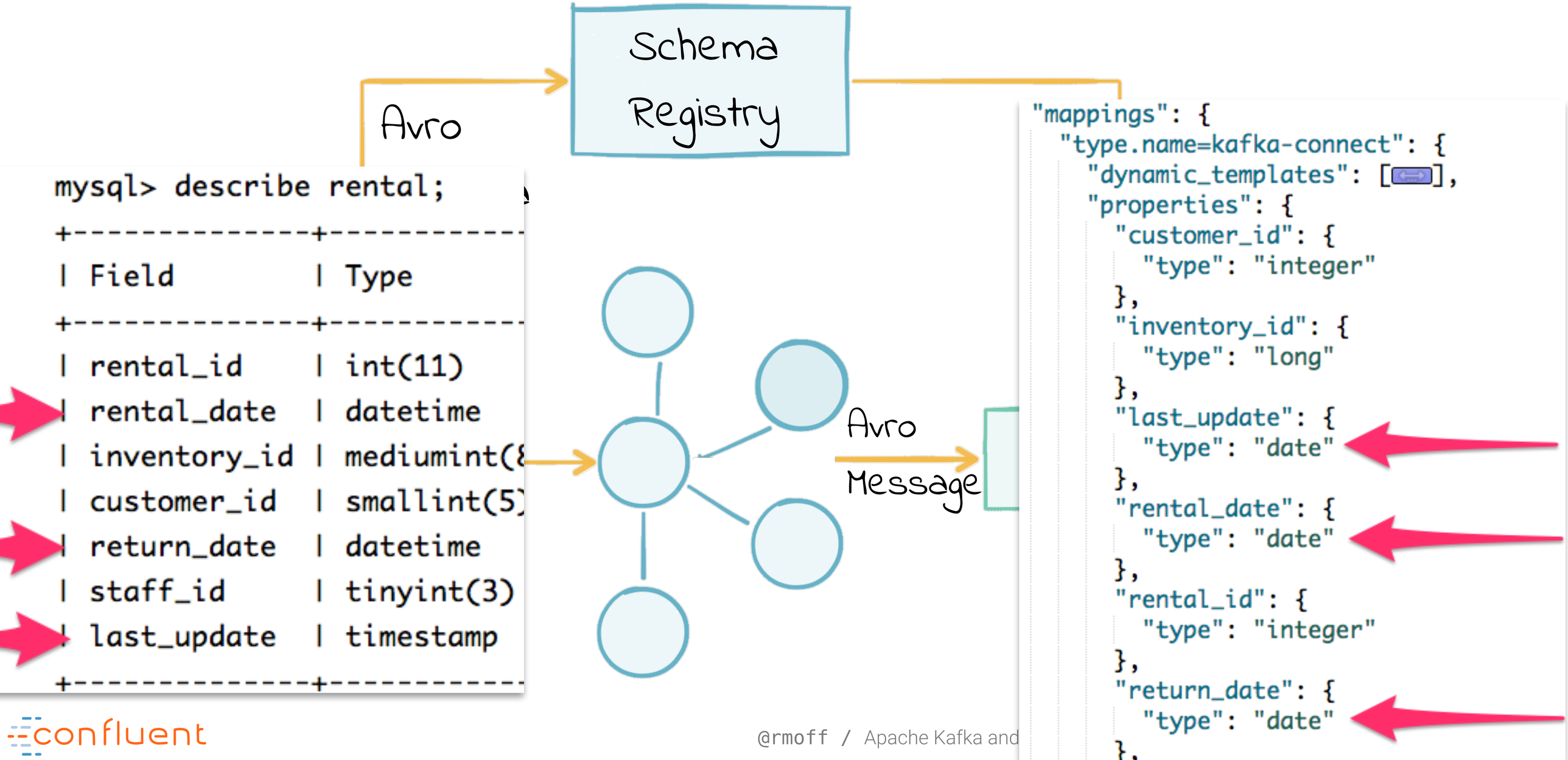
```
{
"connector.class": "io.confluent.connect.jdbc.JdbcSourceConnector",
"connection.url": "jdbc:mysql://localhost:3306/demo?user=rmoff&password=foo",
"table.whitelist": "sales,orders,customers"
}
```

https://docs.confluent.io/current/connect/

Kafka Connect + Schema Registry = WIN

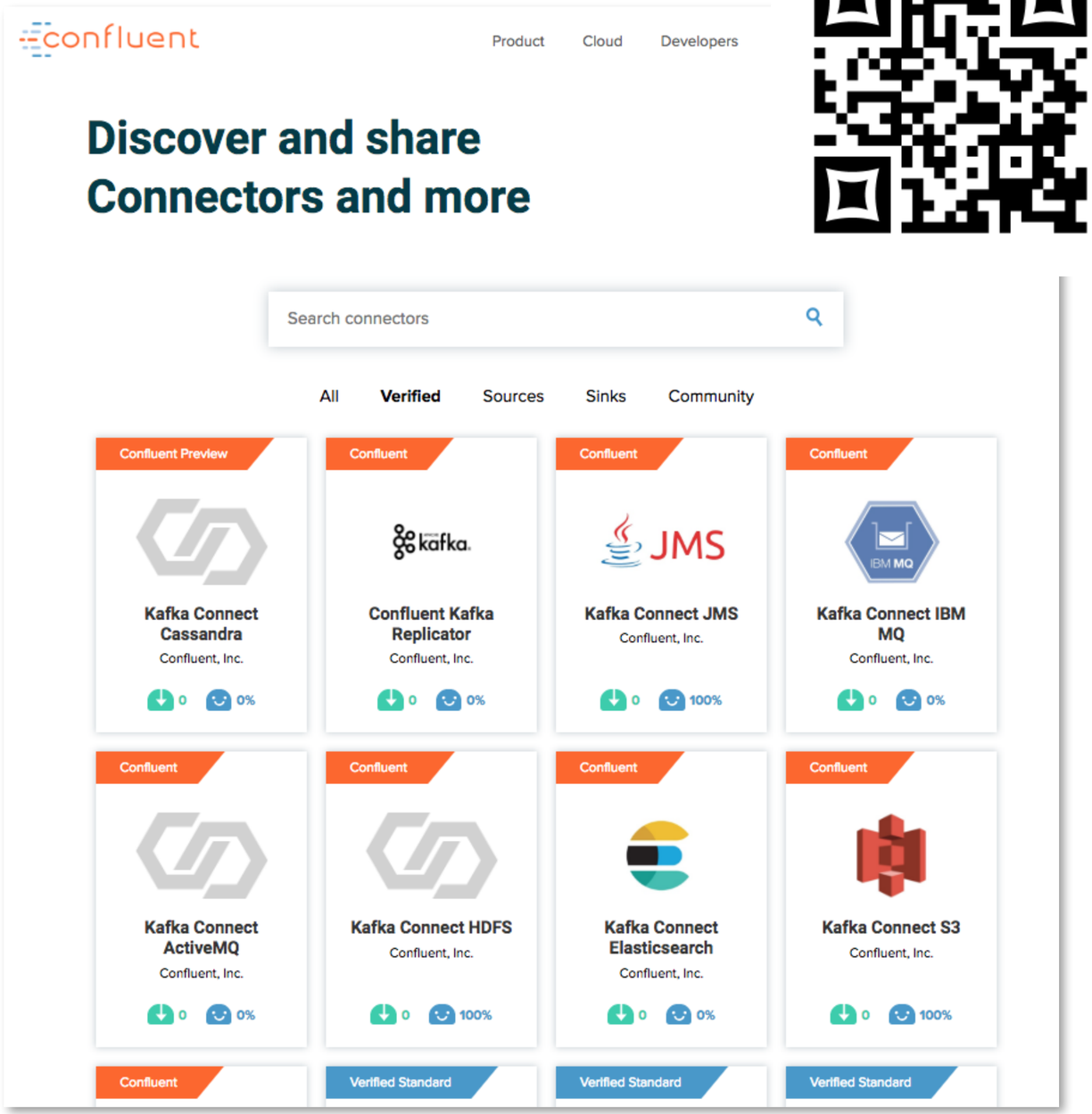


Kafka Connect + Schema Registry = WIN



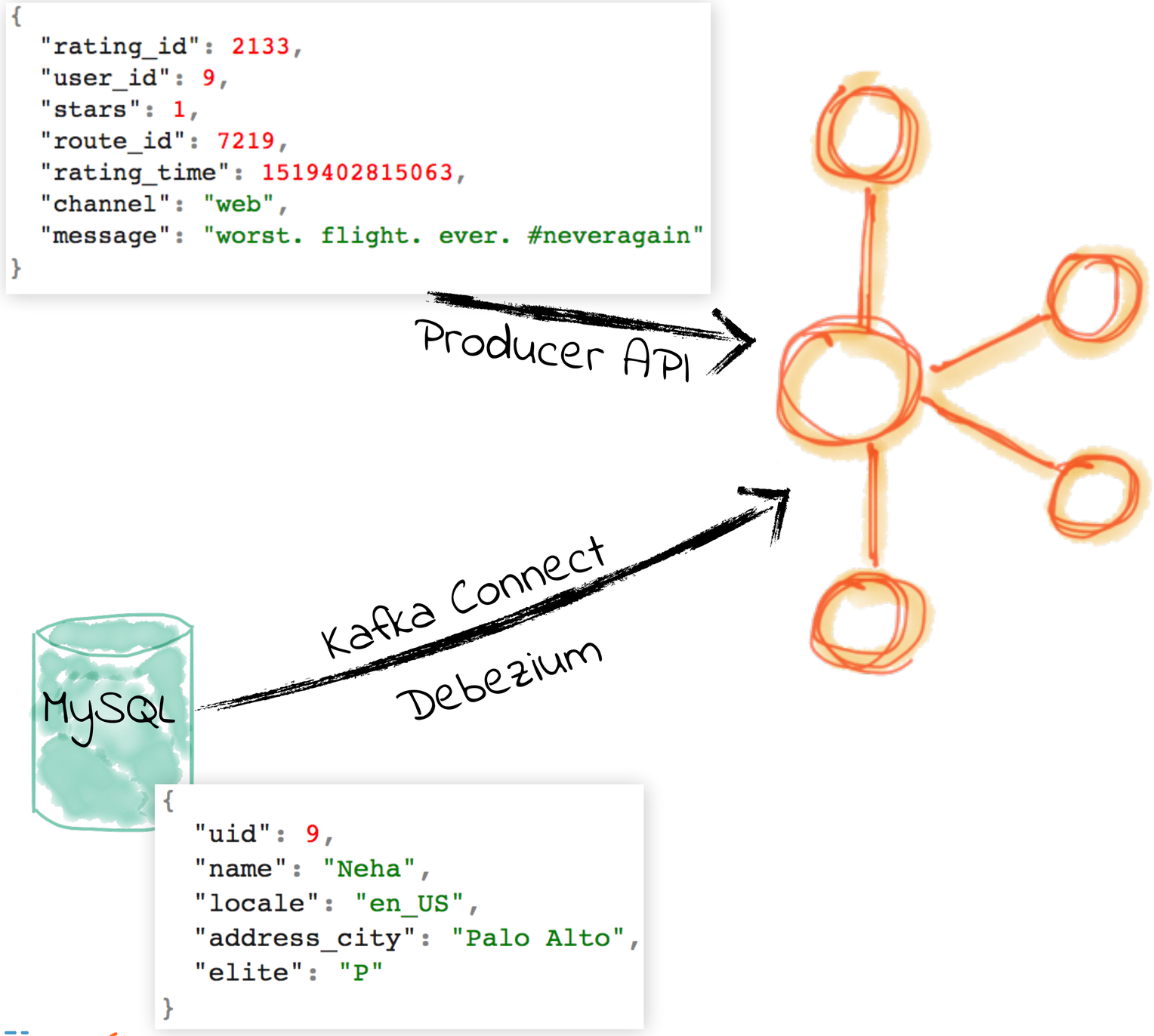
Confluent Hub

- One-stop place to discover and download :
  - Connectors
  - Transformations
  - Converters

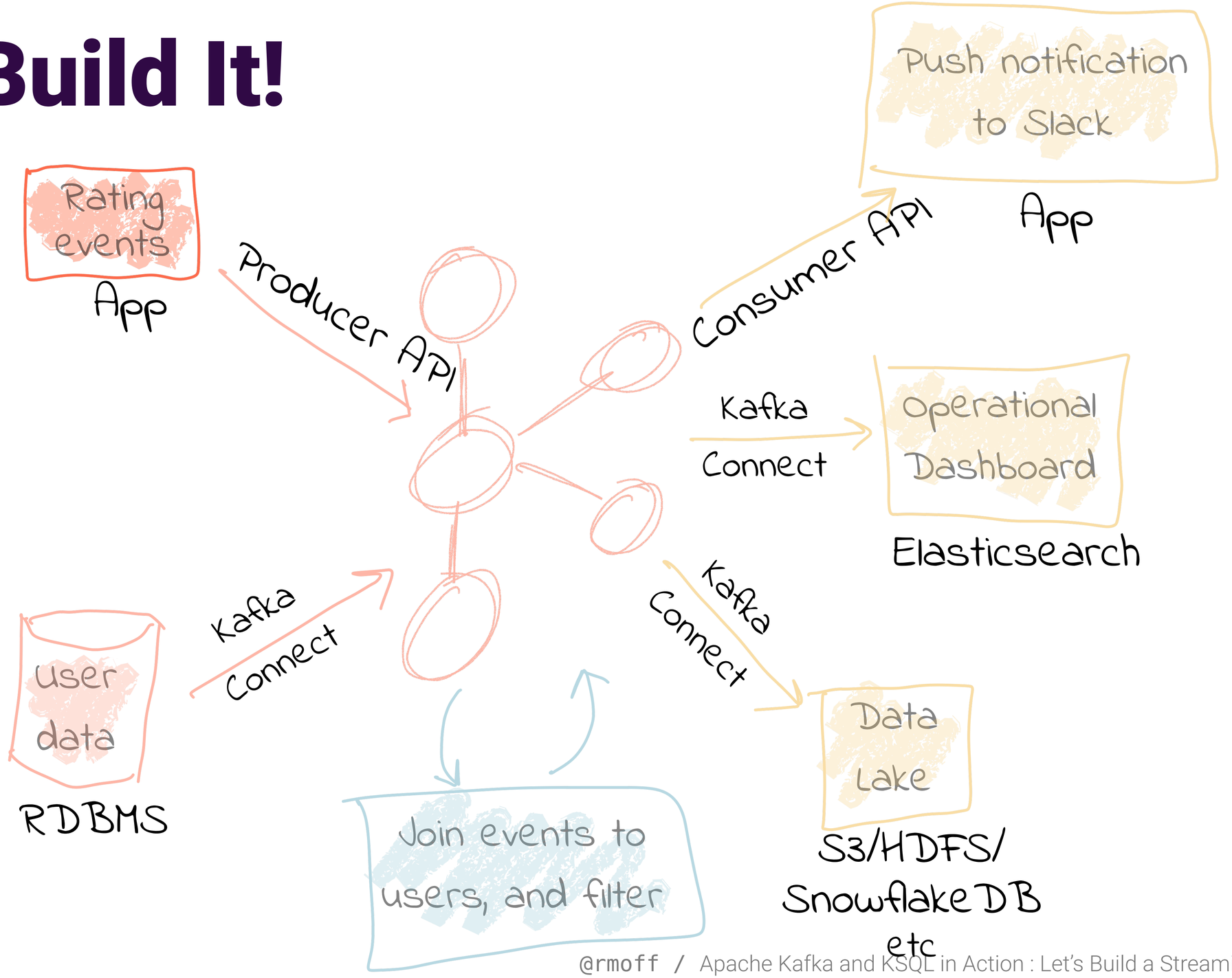


hub.confluent.io

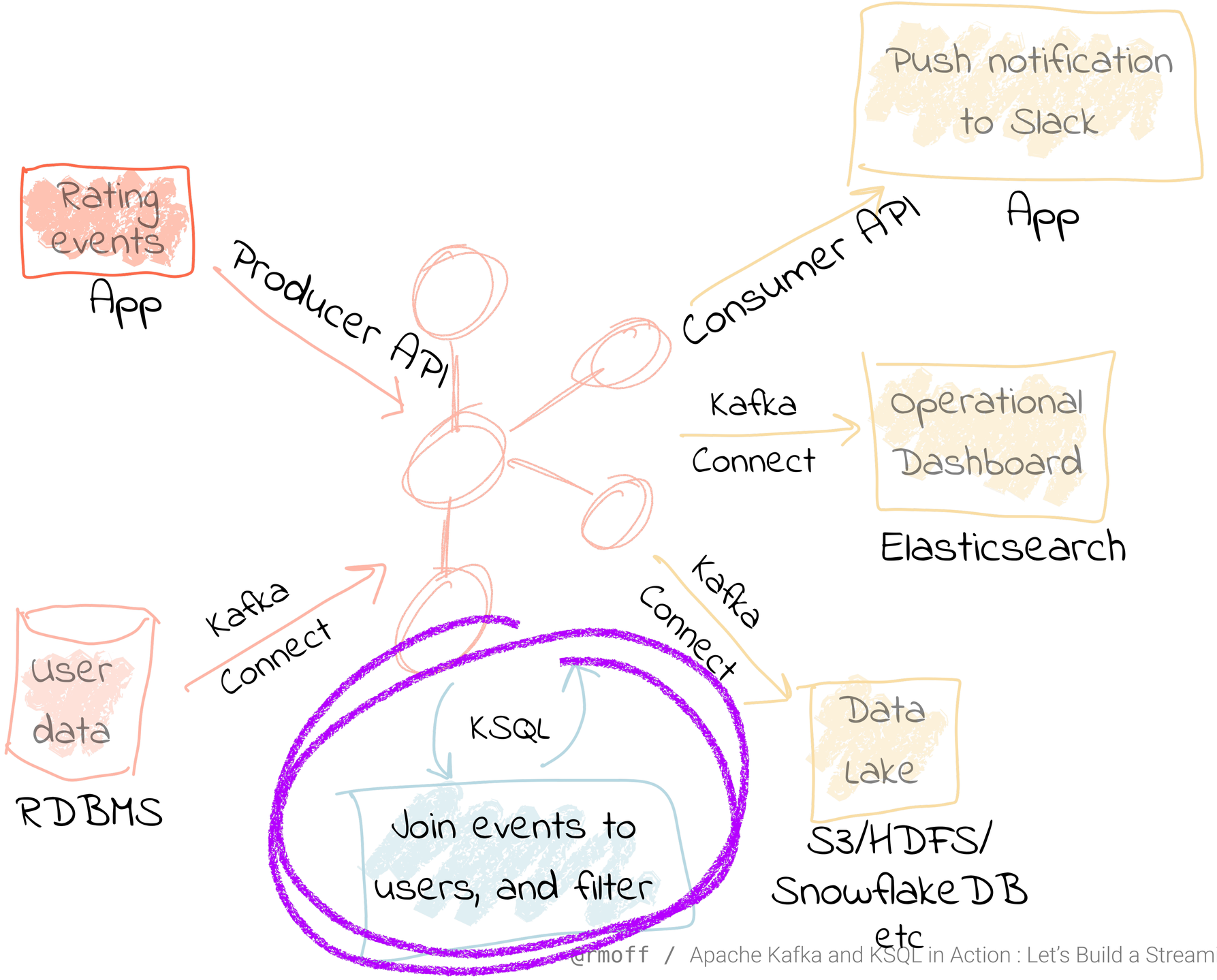
Demo Time!


{
"rating_id": 2133,
"user_id": 9,
"stars": 1,
"route_id": 7219,
"rating_time": 1519402815063,
"channel": "web",
"message": "worst. flight. ever. #neveragain"
}
Producer API
Kafka Connect
Debezium
MySQL
{
"uid": 9,
"name": "Neha",
"locale": "en_US",
"address_city": "Palo Alto",
"elite": "P"
}

Let's Build It!



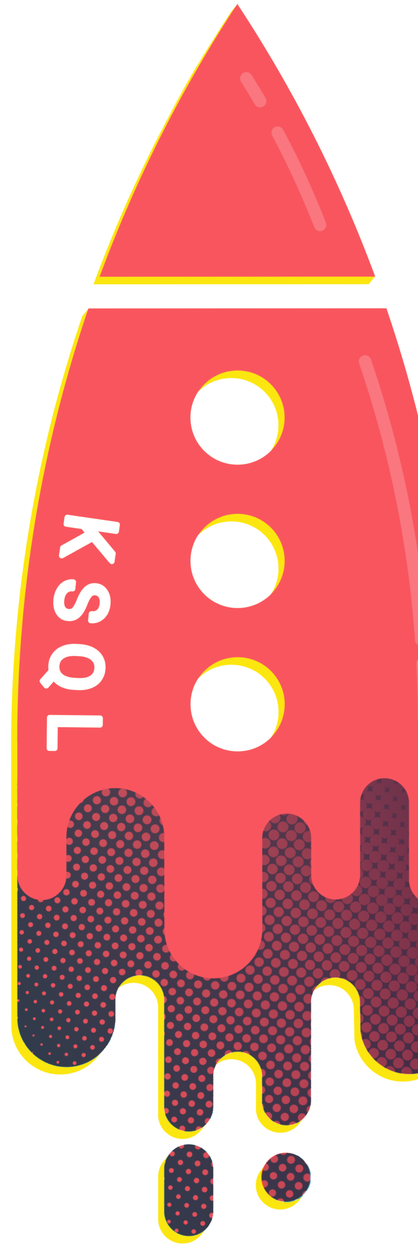
KSQL


Rating events
App
Producer API
Push notification to Slack
App
Consumer API
Kafka Connect
Operational Dashboard
Elasticsearch
user data
RDBMS
Kafka Connect
Kafka Connect
KSQL
Join events to users, and filter
Data Lake
S3/HDFS/ SnowflakeDB etc

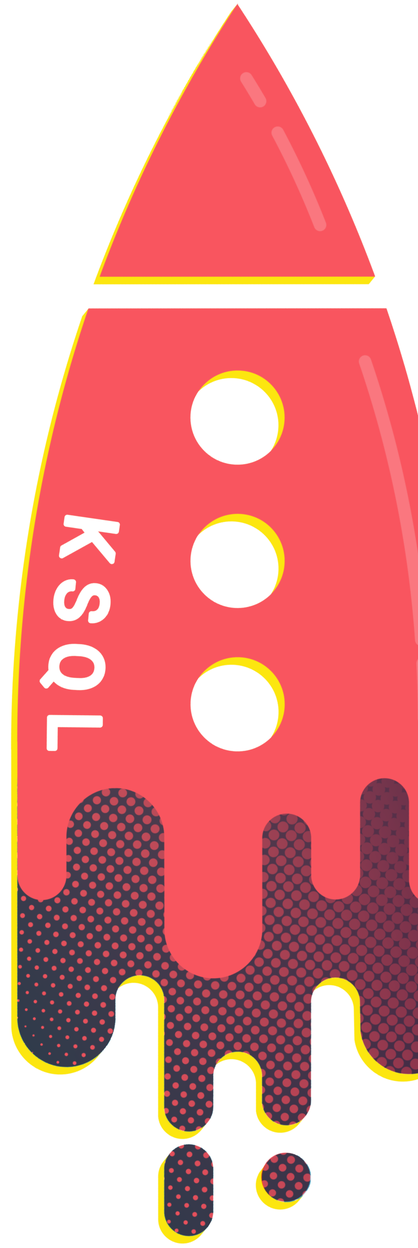
KSQL

is a

Declarative Stream Processing Language



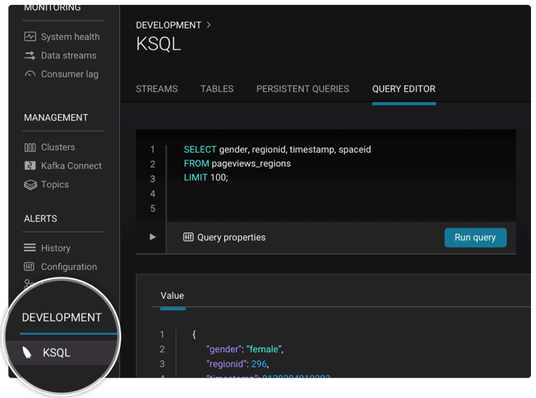
KSQL
is the
Streaming
SQL Engine
for
Apache Kafka



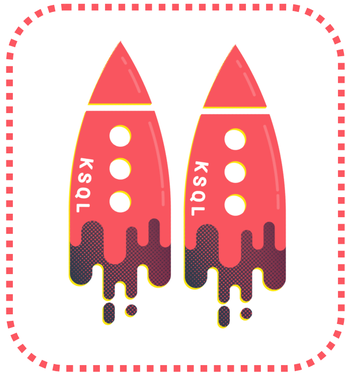
KSQL in Development and Production

Interactive KSQL
for development and testing

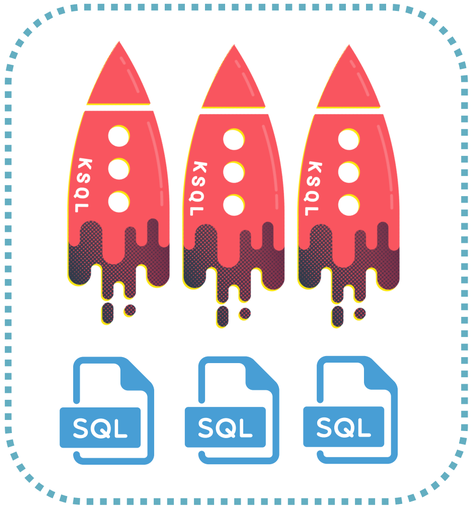
Headless KSQL
for Production



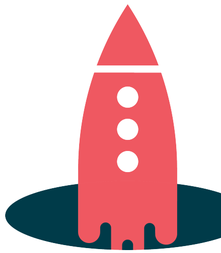
REST



Desired KSQL queries have been identified



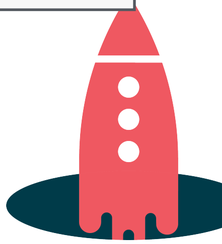
"Hmm, let me try out this idea..."



KSQL for Streaming ETL

Joining, filtering, and aggregating streams of event data

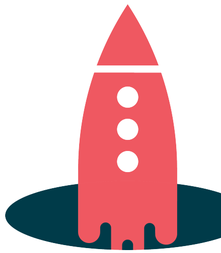
```
CREATE STREAM vip_actions AS
  SELECT userid, page, action
  FROM clickstream c
  LEFT JOIN users u
    ON c.userid = u.user_id
  WHERE u.level = 'Platinum';
```



KSQL for Anomaly Detection

Identifying patterns or anomalies in real-time data, surfaced in milliseconds

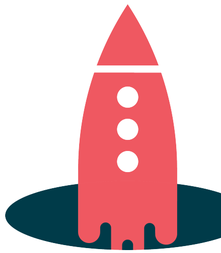
```
CREATE TABLE possible_fraud AS
  SELECT card_number, count(*)
    FROM authorization_attempts
    WINDOW TUMBLING (SIZE 5 SECONDS)
    GROUP BY card_number
    HAVING count(*) > 3;
```



KSQL for Real-Time Monitoring

- Log data monitoring, tracking and alerting
- syslog data
- Sensor / IoT data

```
CREATE STREAM SYSLOG_INVALID_USERS AS
SELECT HOST, MESSAGE
FROM SYSLOG
WHERE MESSAGE LIKE '%Invalid user%';
```

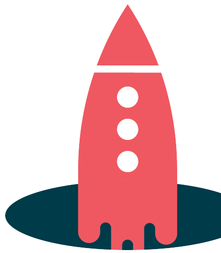


http://cnfl.io/syslogs-filtering / http://cnfl.io/syslog-alerting

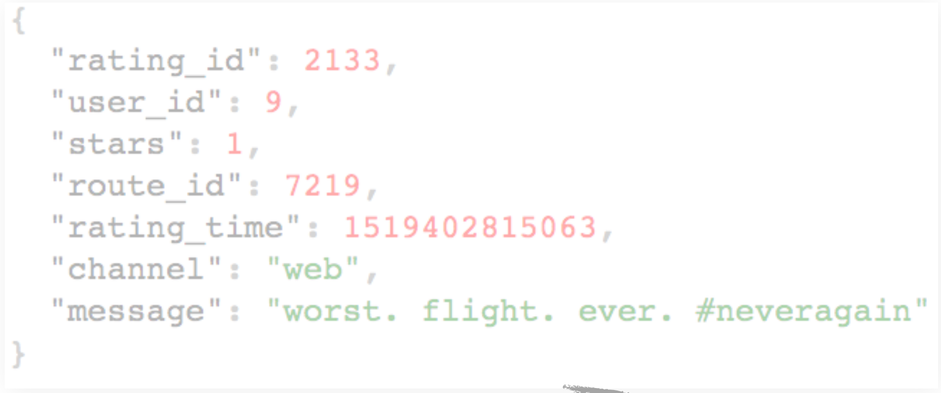
KSQL for Data Transformation

Make simple derivations of existing topics from the command line

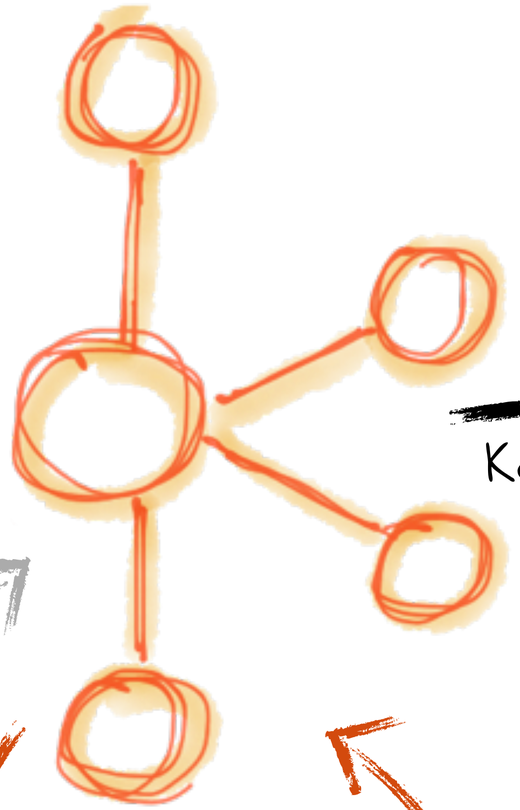
```
CREATE STREAM views_by_userid
  WITH (PARTITIONS=6, REPLICAS=5,
        VALUE_FORMAT='AVRO',
        TIMESTAMP='view_time') AS
  SELECT * FROM clickstream
  PARTITION BY user_id;
```



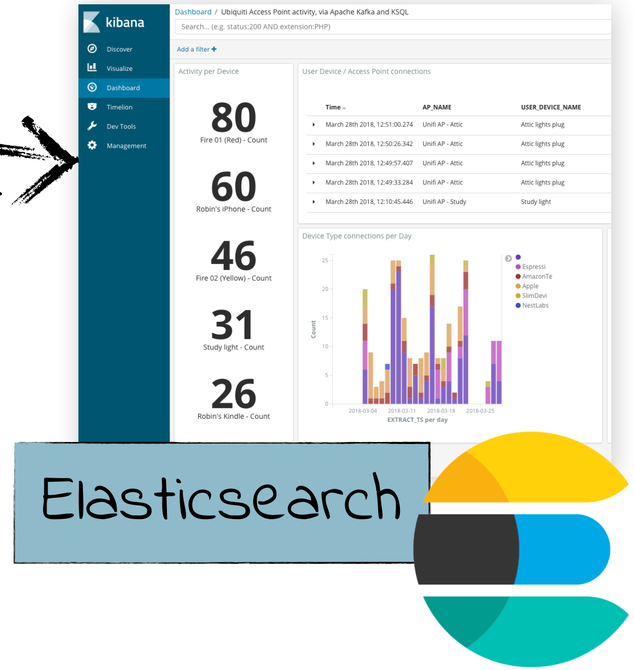
Demo Time!


{
"rating_id": 2133,
"user_id": 9,
"stars": 1,
"route_id": 7219,
"rating_time": 1519402815063,
"channel": "web",
"message": "worst. flight. ever. #neveragain"
}

Producer API

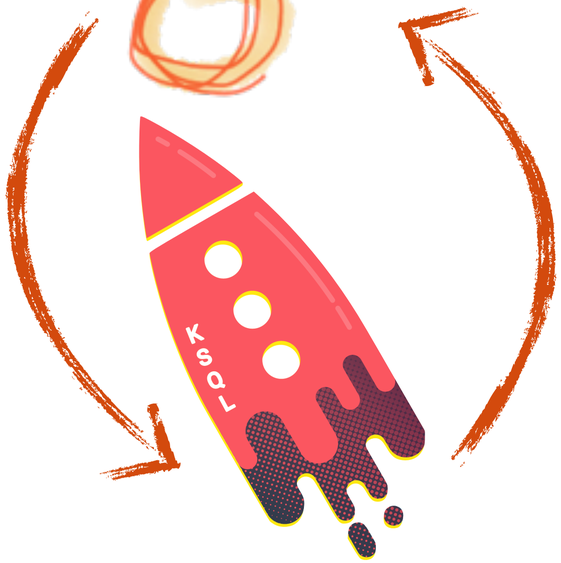


Kafka Connect


kibana
Dashboard / Ubiquiti Access Point activity, via Apache Kafka and KSQL
Search... (e.g. status:200 AND extension:PHP)
Add a filter +
Discover
Visualize
Dashboard
Timelion
Dev Tools
Management
Activity per Device
80
Fire 01 (Red) - Count
60
Robin's iPhone - Count
46
Fire 02 (Yellow) - Count
31
Study light - Count
26
Robin's Kindle - Count
User Device / Access Point connections
Time
AP_NAME
USER_DEVICE_NAME
March 28th 2018, 12:51:00.274 Unifi AP - Attic Attic lights plug
March 28th 2018, 12:50:26.342 Unifi AP - Attic Attic lights plug
March 28th 2018, 12:49:57.407 Unifi AP - Attic Attic lights plug
March 28th 2018, 12:49:33.284 Unifi AP - Attic Attic lights plug
March 28th 2018, 12:10:45.446 Unifi AP - Study Study light
Device Type connections per Day
Count
EXTRACT_TS per day
Elasticsearch

MySQL

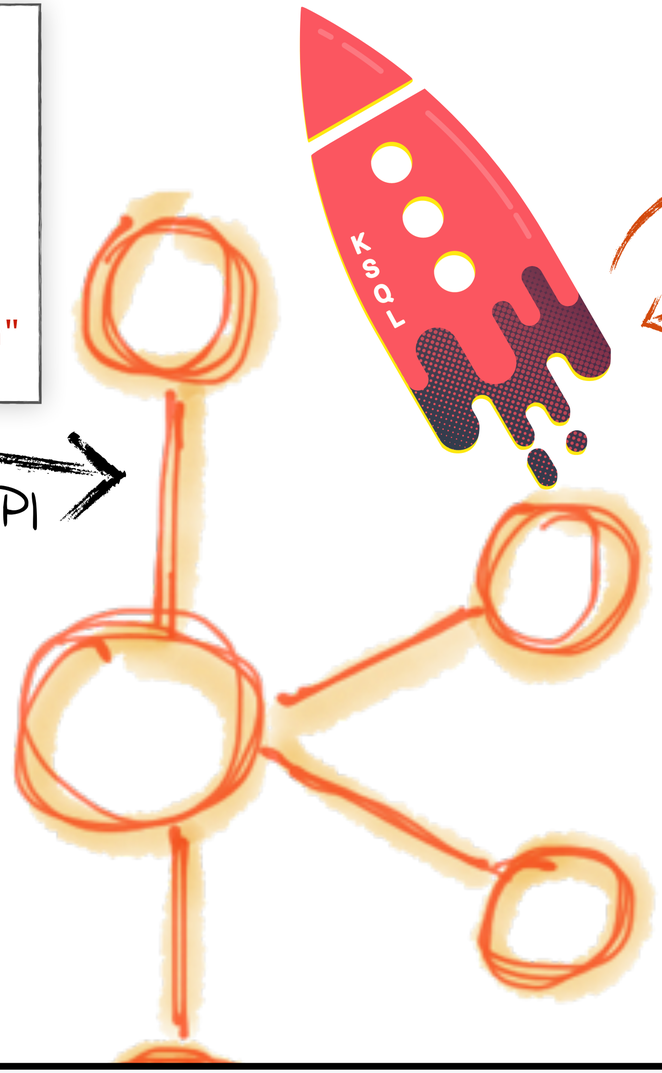
Kafka Connect
Debezium


KSQL

{
"uid": 9,
"name": "Neha",
"locale": "en_US",
"address_city": "Palo Alto",
"elite": "P"
}

```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}
```

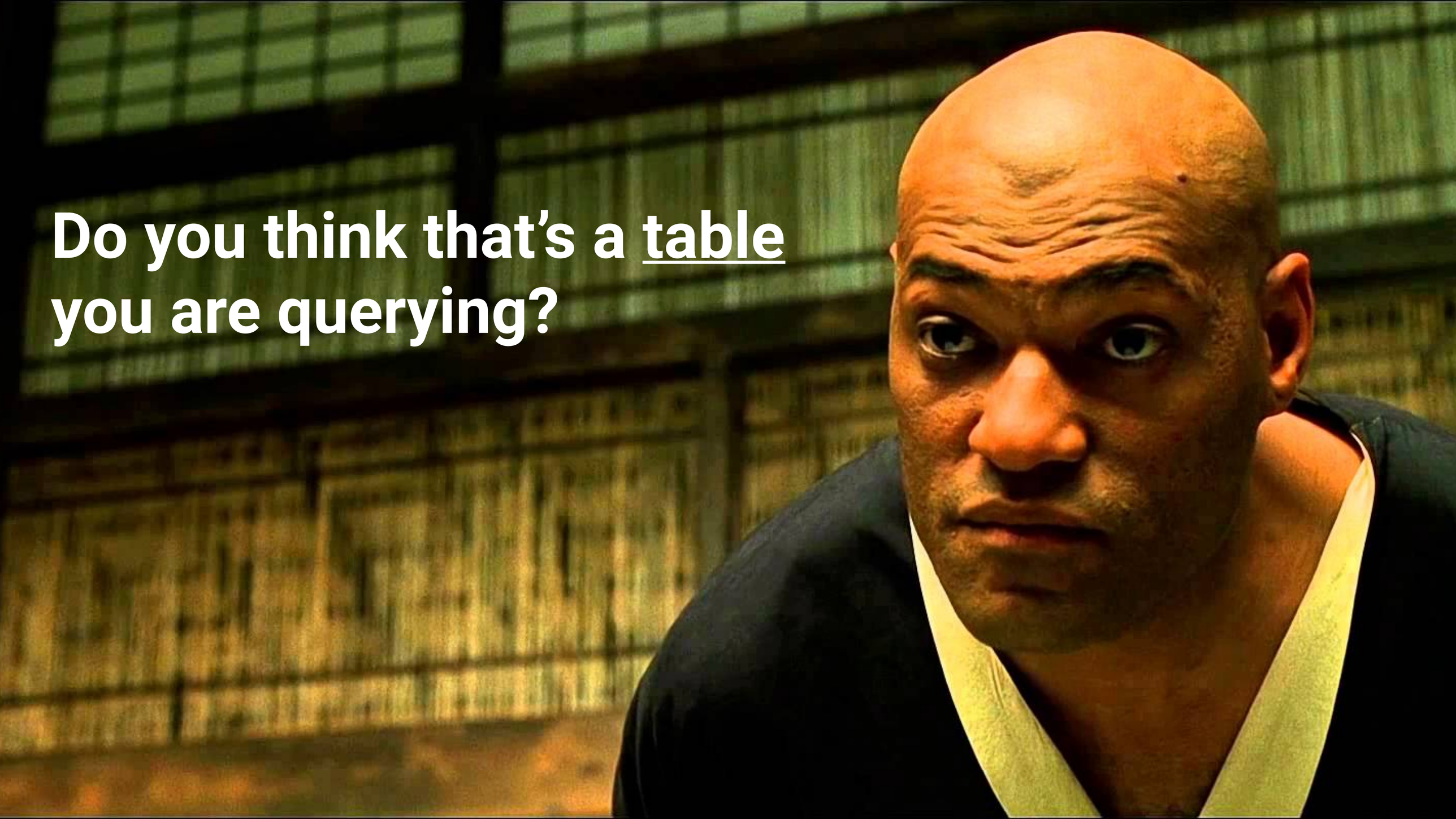
Producer API



Filter all ratings where STARS<3

POOR_RATINGS

```
CREATE STREAM POOR_RATINGS AS
SELECT * FROM ratings WHERE STARS <3
```


Do you think that’s a table
you are querying?

The Table Stream Duality

Stream

Time

Account ID	Amount
12345	+ €50
12345	+ €25
12345	-€60

Table

Account ID	Balance
12345	€50

Account ID	Balance
12345	€75

Account ID	Balance
12345	€15

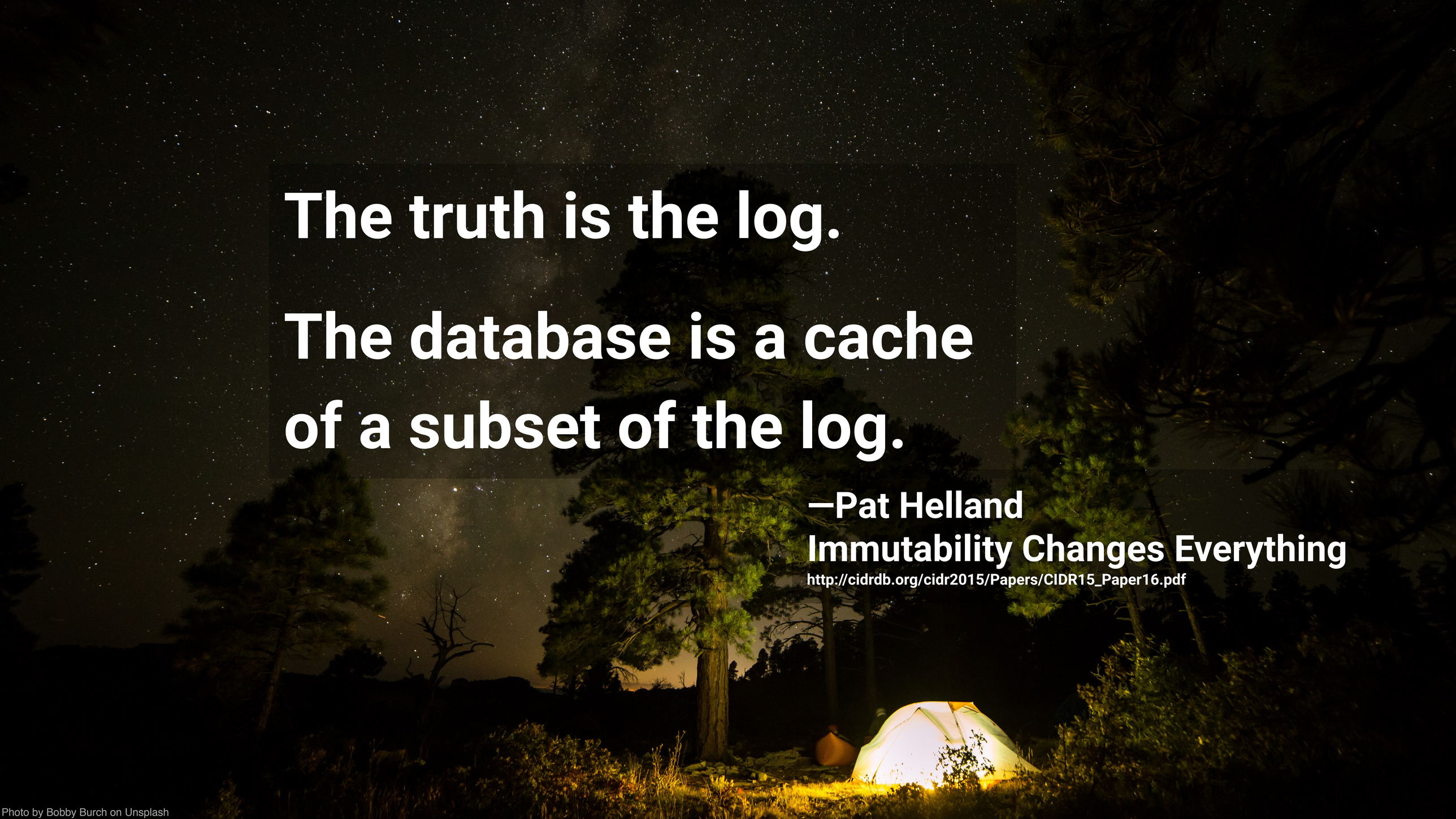
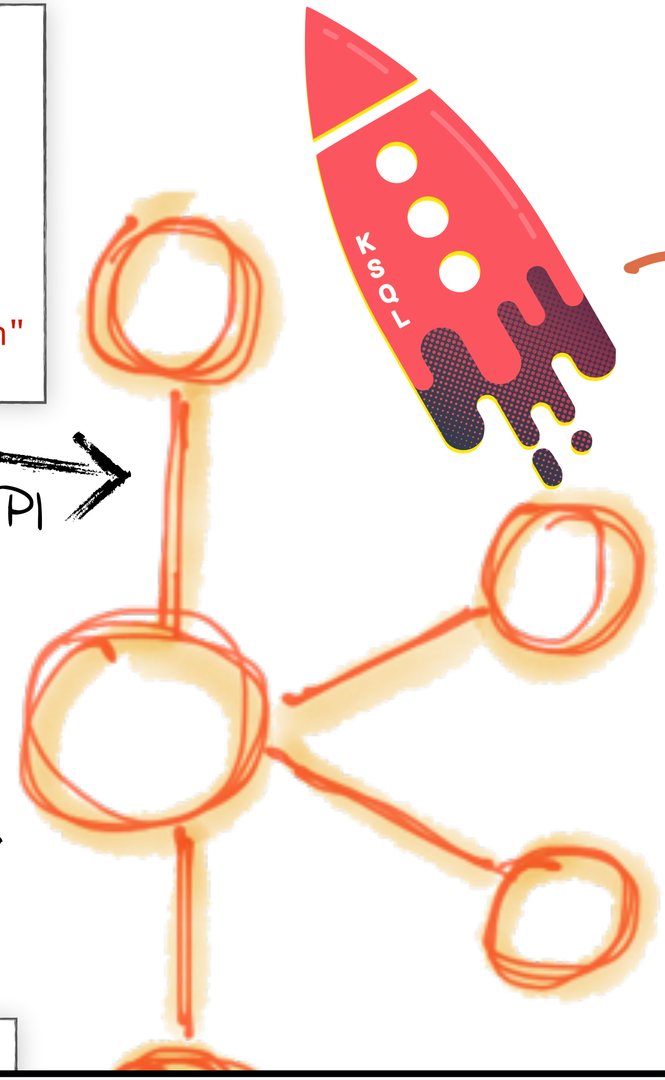


Photo by Bobby Burch on Unsplash

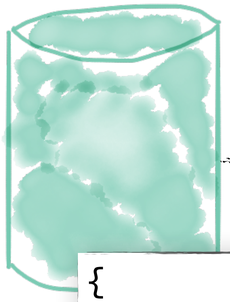
```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}
```


Producer API



Join each rating to customer data

RATINGS_WITH_CUSTOMER_DATA



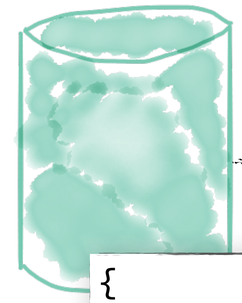
Kafka Connect


```
{
  "id": 3,
  "first_name
  "last_name"
  "email": "m
  "gender": "
  "club_statu
  "comments":
}
```

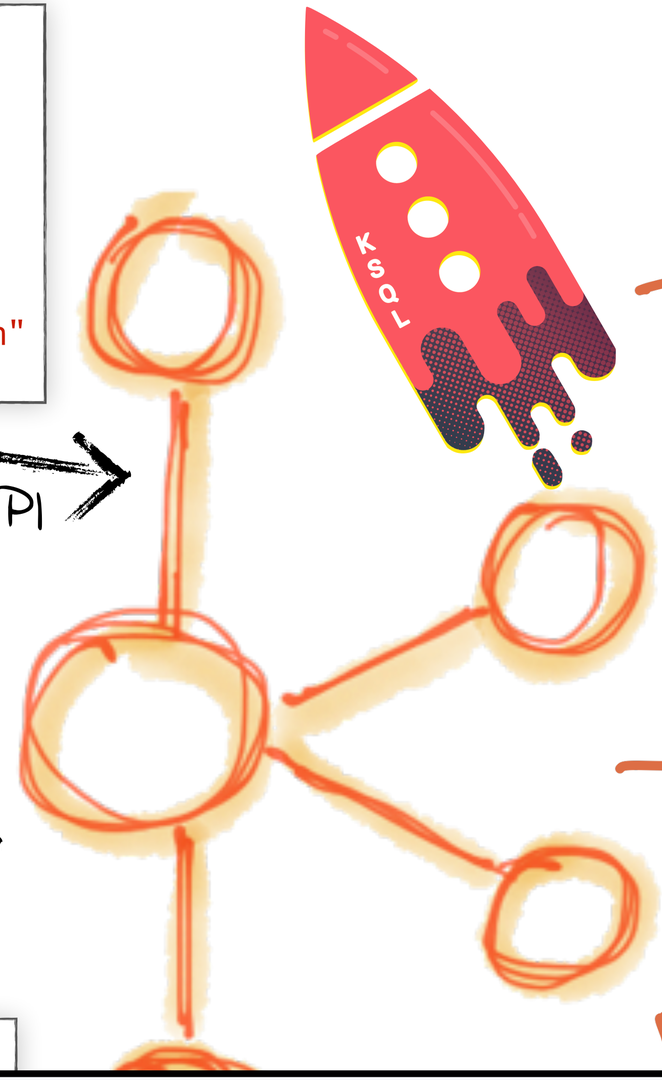

```
CREATE STREAM RATINGS_WITH_CUSTOMER_DATA AS
SELECT * FROM RATINGS LEFT JOIN CUSTOMERS
ON R.ID=C.ID;
```

```
{
 "rating_id": 5313,
 "user_id": 3,
 "stars": 4,
 "route_id": 6975,
 "rating_time": 1519304105213,
 "channel": "web",
 "message": "worst. flight. ever. #neveragain"
}
```


Producer API



Kafka Connect



Join each rating to customer data

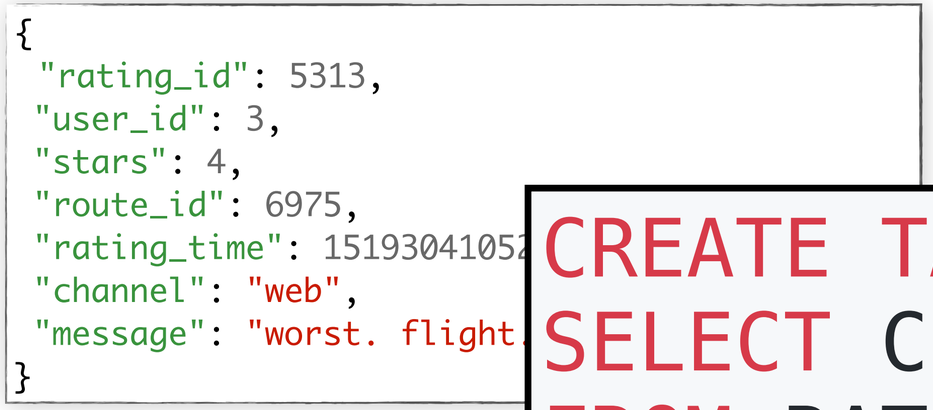
RATINGS_WITH_CUSTOMER_DATA

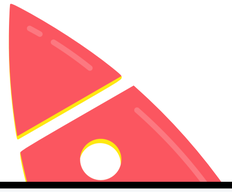
Filter for just PLATINUM customers

UNHAPPY_PLATINUM_CUSTOMERS

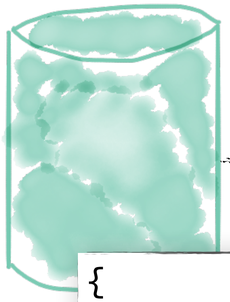

```
{
  "id": 3,
  "first_name
  "last_name"
  "email": "m
  "gender": "
  "club_statu
  "comments":
}
```


```
CREATE STREAM UNHAPPY_PLATINUM_CUSTOMERS AS
SELECT * FROM RATINGS_WITH_CUSTOMER_DATA
WHERE   STARS < 3
```

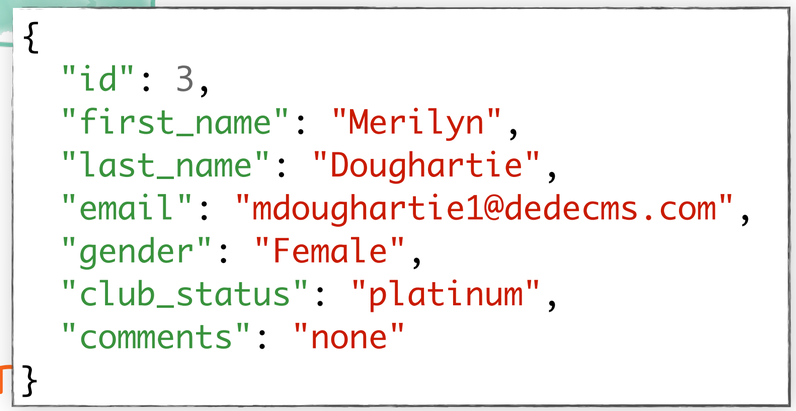




```
CREATE TABLE RATINGS_BY_CLUB_STATUS AS
SELECT CLUB_STATUS, COUNT(*)
FROM RATINGS_WITH_CUSTOMER_DATA
     WINDOW TUMBLING (SIZE 1 MINUTES)
GROUP BY CLUB_STATUS;
```



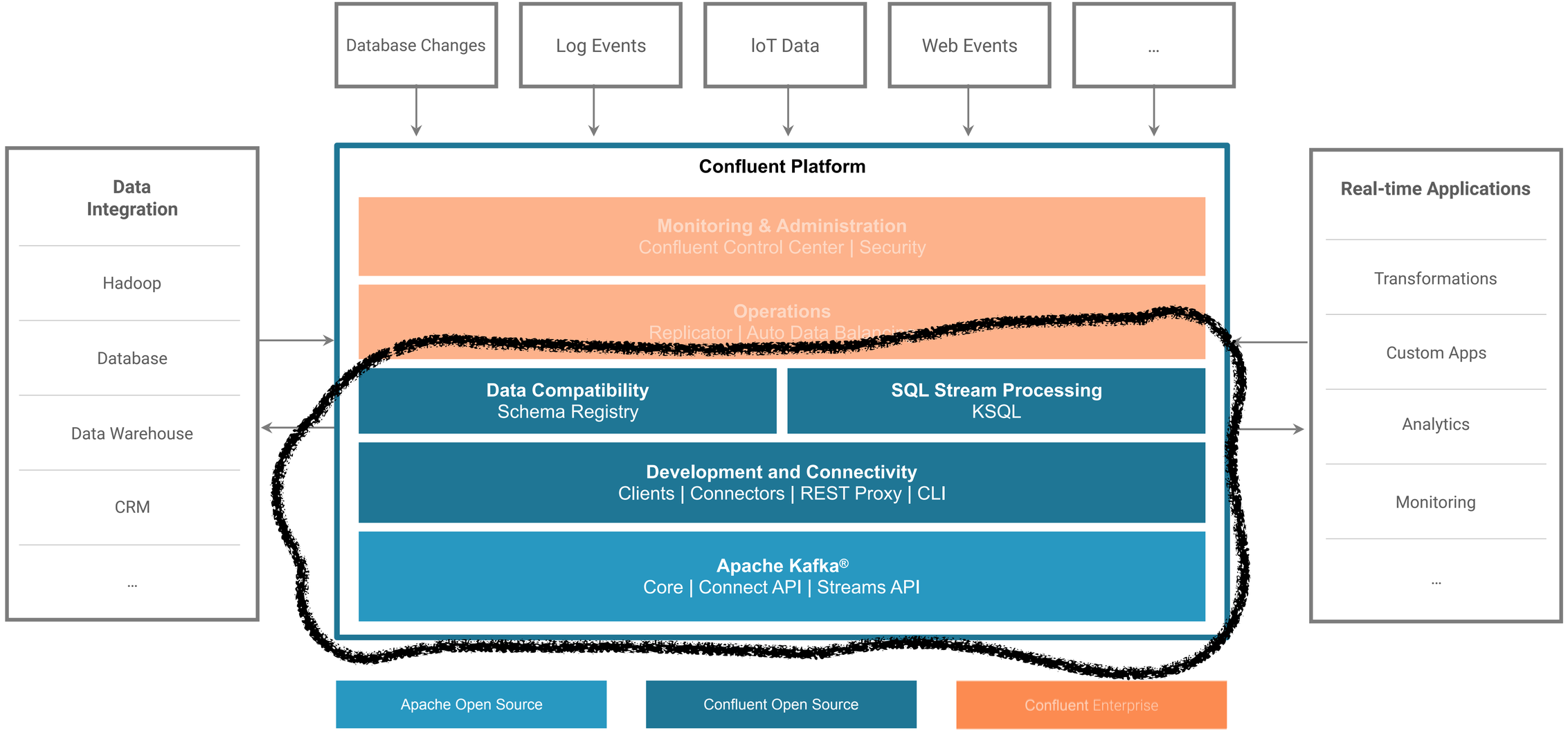
Kafka Connect



Aggregate per-minute by CLUB_STATUS

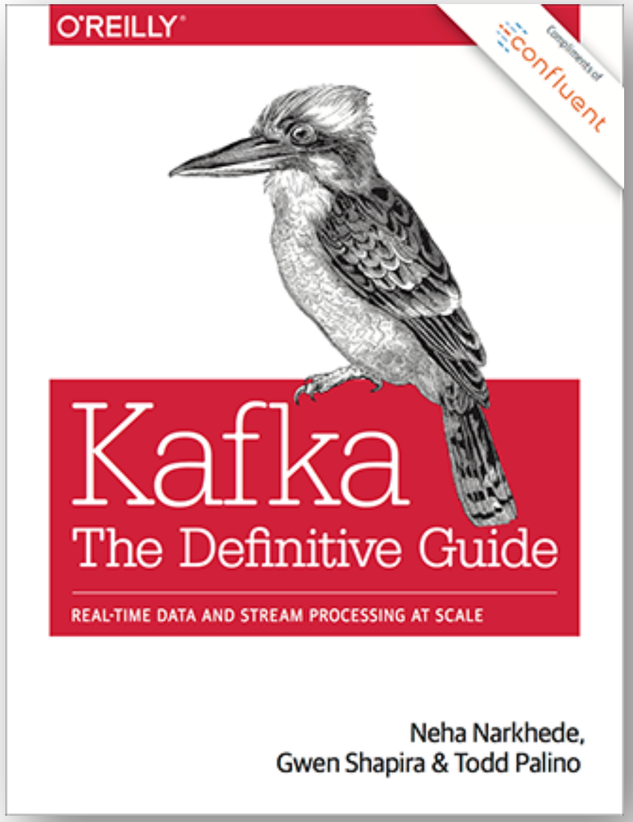
RATINGS_BY_CLUB_STATUS_1MIN

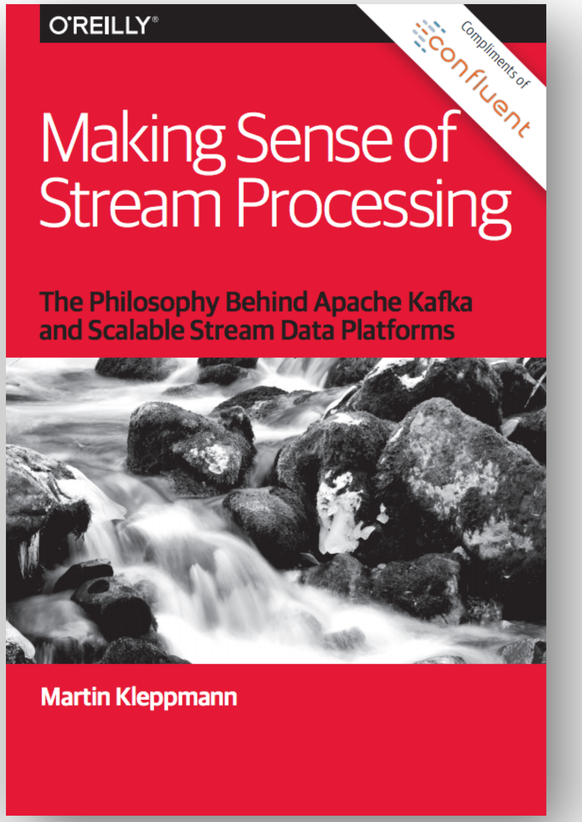
Confluent Open Source : Apache Kafka with a bunch of cool stuff! For free!

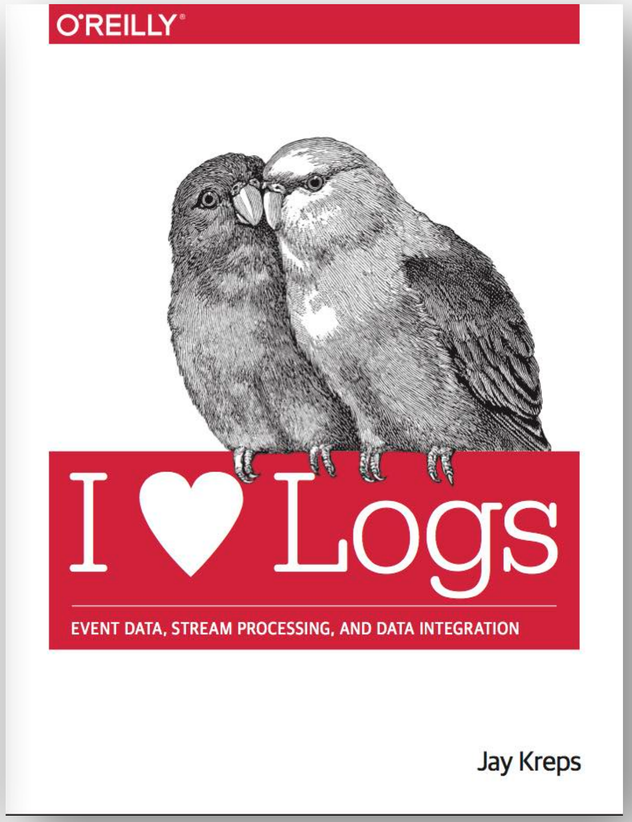


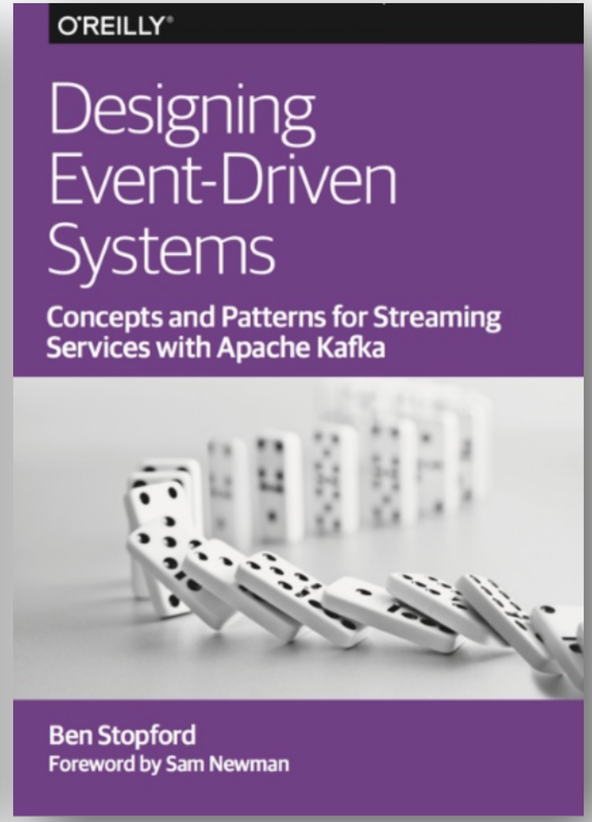
Free Books!

https://www.confluent.io/apache-kafka-stream-processing-book-bundle









https://www.confluent.io/download/
http://cnfl.io/slack
@rmoff
robin@confluent.io

Useful links

- Embrace the Anarchy : Apache Kafka's Role in Modern Data Architectures Recording & Slides
- Look Ma, no Code! Building Streaming Data Pipelines with Apache Kafka and KSQL
- Steps to Building a Streaming ETL Pipeline with Apache Kafka and KSQL Recording & Slides
- https://www.confluent.io/blog/ksql-in-action-real-time-streaming-etl-from-oracle-transactional-data
- https://github.com/confluentinc/ksql/

#EOF